A VOXELIZED FRAMEWORK FOR SIMULATING CUTTING TOOL AND WORKPIECE INTERACTION

A Dissertation
Presented to
The Academic Faculty

by

John Carter Miers

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2019

A VOXELIZED FRAMEWORK FOR SIMULATING CUTTING TOOL AND WORKPIECE INTERACTION

Approved by:

Dr. Christopher Saldana, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Thomas Kurfess
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Thomas Tucker
Principal of Tucker Innovations
Tucker Innovations Inc.

Date Approved: November 28, 2018

To my wife Savannah Miers, for all the love and support during this process.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my committee members, Dr. Thomas Kurfess and Dr. Tommy Tucker for their time and wisdom help in the process of my master's research. I would also like to thank Mrs. Glenda Johnson for her patience and always valued help in the finishing of my master's degree here at Georgia Tech. The conversations and advice I have received over the years have helped me clarify what questions I should ask, how I should go about searching for the answers, and providing me with solutions, often times designed specifically for my needs.

I would also like to thank my advisor, D. Christopher J. Saldana, for the time and resources he has poured into me to get me to this point. His guidance and thoughts on my research have been invaluable. His thorough research standards and his shared lessens have helped me sharpen my own research and skills in this process. I would like to thank my wife, Savannah Miers, and my lab mates: Haipeng Qiao, Joon Kim, Max Praniewicz, Saurabh Singh, Vinh Nguyen, and Zhiyu Wang; for the inspiration and encouragement along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

i	The index of the x-direction in the workpiece coordinate system.
j	The index of the y-direction in the workpiece coordinate system.
k	The index of the z-direction in the workpiece coordinate system.
w	The variable of the isotropic width of each voxel.
ω	The spindle speed of the cutting tool.
D	The diameter of the cutting tool at the plane normal to the tool axis and located at the center point of the tool.
O	The point at the tip of the tip of the tool in the tool envelope.
M	The point at the beginning of the radii portion of the tool envelope when traveling from O.
N	The point at the end of the radii portion of the tool envelope when traveling from M.
G	The point at the top or end of the tool flank or at the end of the flank portion of the envelope when travelling from N.
L	The point projected coincident with both lines OM and MN.
M_r	The distance of point M from the tip, O, measured orthogonal to the tool's central axis in the tool envelope.
M_z	The distance of point Malong the tool axis measured from the tip, O, of the tool in the tool envelope.
N_r	The distance of point N from the tip, O, measured orthogonal to the tool's central axis in the tool envelope.

N_z	The distance of point Nalong the tool axis measured from the tip, O, of the tool in the tool envelope.
G_r	The distance of point G from the tip, O, measured orthogonal to the tool's central axis in the tool envelope.
G_z	The distance of point Galong the tool axis measured from the tip, O, of the tool in the tool envelope.
L_r	The distance of point L from the tip, O, measured orthogonal to the tool's central axis in the tool envelope.
L_z	The distance of point L along the tool axis measured from the tip, O, of the tool in the tool envelope.
R	The radius of the radii portion, MN, of the tool envelope.
R_r	The distance of the center of curvature of the radii portion of the tool envelope measured from the tip, O, orthogonal to the tool's axis.
R_z	The distance of the center of curvature of the radii portion of the tool envelope along the tool axis measured from the tip, O.
h	The total height of the tools cutting portion measured from the tip, O, along the tools axis in the tool envelope.
α	The angle of the tip region, OM, of the tool envelope measured from the horizontal axis.
β	The angle of the flank region,NG, of the tool envelope measured from the tool's axis.
ds	The arc length between consecutive rake lines along the flute of the generated tool.
φ_{o,R_ζ}	The angles of the current discrete point along section $OM, MN, \& NG$ of the cutting tool envelope. Measured from the tool axis.
ξ	The index of the discrete steps in time.
Γ	The index of the specific cutting flutes.
ζ	The index of the discrete rake lines.

r	The radius of the discrete points of the outer most edge of the cutting flutes from the tool axis.
z	The height of the discrete points of the outer most edge of the cutting flutes from the tool axis.
κ	The angle of the discrete points of the outer most edge of the cutting flutes from the tool axis.
γ	The angular displacements of the discrete points of outer most edge of the cutting flutes measured around the tool's axis.
τ_{total}	The total time of the simulation in seconds.
$\Phi_{referenceflute}$	The angle of each discrete point of the outer most edge of the first cutting flute measured around the tool's axis.
Φ_{flutes}	The angle of each discrete point of the outer most edge of all the cutting flutes measured around the tool's axis.
$\vec{e}_{feedrate}$	The unit vector or series of unit vectors describing the direction of the tools travel in the global coordinate system.
$Feedrate$	The value of the tools feedrate in distance/min.
$P.tip$	The location of the tip of the tool at every temporal step of the simulation.
C	The starting point of the tool's tip in the global coordinate system.
N_{flutes}	The number of cutting flutes for the specified tool.
$\overrightarrow{e_i}$	The unit vector of the x direction in the global coordinate system.
$\overrightarrow{e_j}$	The unit vector of the y direction in the global coordinate system.
$\overrightarrow{e_k}$	The unit vector of the z direction in the global coordinate system.

$P.coord$	An array that contains the location of the all the discrete points of the outer most edge of the cutting flutes.
L_{rake}	The length of the rake lines on the generated cutting tool.
α_{rake}	The angle of the rake face with respect the radial line between the tool axis and the corresponding rake face's discrete point on the outer most edge of the tool's cutting flutes.
t_{axis}	An offset of the tool axis used to calculate the each rake line.
$P.rake$	An array that contains the location of the all the discrete inner points of the rake lines of the cutting flutes.
$\mu, \eta, \& \lambda$	The angles of rotation in the a-, b-, & c-axis corresponding to rotations about the x-, y-, & z-axis respectively.
τ_{ξ}	A specific time at temporal step ξ.
$\boldsymbol{R}_{\xi}$	The tool orientation rotation matrix at temporal step ξ.
$S_{min}(q)$	The distance between the starting point of a ray and the minimum bound of the workpiece boundary for a given component, q, in the global coordinate system.
$S_{max}(q)$	The distance between the starting point of a ray and the maximum bound of the workpiece boundary for a given component, q, of the direction in the global coordinate system.
q	A dummy variable that represents the component directions for the global coordinate system.
$Bound_{min}\ (q)$	The location of the workpieces minimum boundary for a given component, q, in the global coordinate system.
$\underline{\hat{d}}(q)$	The value of the component, q, of the unit vector or direction of a ray in the global coordinate system.
$\boldsymbol{p}_{start}$	The floating point location of the first point along a ray to penetrate a voxel of the workpiece volume.

$x_i, y_j, \& \, z_k$	The indices of the voxels in the $i, j, \& \, k$ component directions of the global coordinate system.
$\Delta x_i, \Delta y_j, \& \, \Delta z_k$	The unit step of the next voxel along the ray direction, where the sign of the component is assigned from the sign of the components of the ray's direction.
$\boldsymbol{p}_{next}$	The floating point location in the global coordinate system of the next voxel the ray travels to.
dS	The minimum distance the ray must travel in each component direction to reach the next voxel.
ΔdS	The distance in each component direction added to the value of a specific component dS each time the ray travels in the direction of that component.
$condition$	The conditions for the termination while loop at the end of the voxel traversal for each ray.
$H_{\xi,\Gamma,\zeta}$	The thickness of the undeformed chip removed by the rake line, ζ, at temporal step, ξ, for flute γ.
$N_{x_i}, N_{y_j}, \& \, N_{z_k}$	The count of how many voxels were traversed in each component direction for the current ray being calculated.
$\begin{pmatrix} dF_{r_{\xi,\Gamma,\zeta}} \\ dF_{t_{\xi,\Gamma,\zeta}} \\ dF_{a_{\xi,\Gamma,\zeta}} \end{pmatrix}$	The differential cutting forces in the radial, tangential, and axial directions of the cutting tool respectively.
$\begin{pmatrix} K_{r_c} \\ K_{t_c} \\ K_{a_c} \end{pmatrix}$	The cutting force coefficients accounting for the shear zone mechanisms of the radial, tangential, and axial directions of the cutting tool respectively.
$\begin{pmatrix} K_{r_e} \\ K_{t_e} \\ K_{a_e} \end{pmatrix}$	The cutting force coefficients accounting for the ploughing mechanisms of the radial, tangential, and axial directions of the cutting tool respectively.
$\boldsymbol{R}_{i,j,k}$	The rotation matrix for defined to transform each differential force of the tool coordinate system into the global coordinate system.

$\begin{pmatrix} dF_{x_\xi} \\ dF_{y_\xi} \\ dF_{z_\xi} \end{pmatrix}$	The differential forces for the global coordinate system at temporal step ξ.

SUMMARY

In this manuscript, a voxel based model for the interaction between cutting teeth of an arbitrary end mill geometry and a workpiece is developed that allows for the virtual machining of workpiece volumes with generated tool geometry. In this framework, the workpiece geometry is modeled using a voxelized representation that is dynamically updated as material is locally removed by each tooth of the cutting tool. A ray casting approach is then used to mimic the process of the cutting faces of the tool raking out workpiece material. This ray casting regime is also used to calculate the instantaneous undeformed chip thickness. The resulting voxel based model framework was validated by comparison of predictions with experimentally measured milling forces. The results demonstrate the model's ability to accurately simulate the interaction of cutting teeth with the bulk material of the workpiece. The model is further expanded to simulate the impact of previous tool passes on subsequent one. Implications of this new voxel based model framework are briefly discussed in terms of utility for predicting local surface finish and computational scalability of complex cutting configurations.

CHAPTER 1. INTRODUCTION

1.1 Background and Motivation

Machining, as the most widely used process in the manufacturing of mechanical components in highly industrialized countries, plays an important role in the development and maintenance of a healthy economy both at home and globally [2]. Machining is vital to a nation being able to produce products and maintain a healthy stream of trade around the world. As recently as 2012 the export of machinery manufactured in the United States topped $400 billion [3], and after several years of returned growth to the global economy machining and machine tools will be ever more important to helping countries maintain their economic growth [4]. Therefore, development of cutting edge machine tools and machining research is ever more important to the sustenance of a healthy economy. As modern manufacturing moves well into the digital era, the application and reliance of computer numerically controlled (CNC) machine tools continues to grow. With these automated machining process complex and highly unique tool paths are easy to implement and can improve the efficiency of the machining process considerably. In addition, as the demand for unique and low production run parts continues to ramp up, competition is growing rapidly. All of these trends combine mean that it is often no longer effective or affordable to allow for time spent with low production due to difficulties in hammering out new the machining of new products.

Now more than ever ‘time is money,’ and as such the era of padding production runs with a period of trial and error optimization is gone. Research is needed to help fill this demand for high precision and low scrap rates for parts. With the implementation of a fully

numerically controlled production floor, companies will be looking to accurate and highly robust machining models that can be used to supplement the lack of familiarity with the optimized path planning of many computer aided machining software packages. Since conventional machining practices suffer from the need to manually select the cutting conditions from prior experience and machining handbooks, it is common for the finalized production runs of components to be performed with highly conservative parameters. It has now become uneconomical to machine complex geometries with these dated practices. Therefore, it is advantageous to be able to predict machining forces and workpiece geometry by simulating problematic or critical areas before actually machining the workpieces. The objective of the current research is to develop a generalized discrete framework for the simulation of the interaction between an arbitrary cutting tool and the body of material being machined.

1.2 Literature Review

The literature reviewed during this research can be broken down into three primary areas: previous works on modeling the machining process, previous works on mathematically representing the geometry of a cutting tool, and previous works on developing efficient and accurate ray casting computer graphics techniques. As M. E. Merchant described the process of modeling the machining process has advanced through three stages: empirical modeling, science-based predictive modeling, and computer- based modeling. The trend of empirical modelling began just before the turn of the 20^{th} century, with predictive modelling following almost a half a century later. Finally, computer-based modelling of the machining process began slowly in the 1970s [2]. Now all three

approaches are used to generate increasingly encompassing models to address the needs of an increasingly competitive industry.

1.2.1 Modelling of the Cutting Process

Unlike the relatively constant chip thickness found in turning processes, in milling the cutting teeth of the tool travel in a trochoidal motion due to the translation of the tool known as its feed and the simultaneous rotation about its axis. In addition, milling is a periodic material removal operation over multiple points along the tool. As such, an individual portion of a cutting tooth is only in contact for a small portion of the period of the tool's rotation. In up-milling operations the rotation of the cutting tool is counter to the feed direction. This means that as the cutting tooth engages with the workpiece material the chip thickness increases. Conversely, down-milling operations involve the chip entering the workpiece at the thickest portion of the chip and exiting at the thinnest. This is due to the rotation of the cutter in down-milling being complementary to the feed direction.

If the period of empirical modelling can be thought as the development of the art of machining by naively experimenting with machining parameters, then the beginnings of the era where engineering was applied to this decision making process came from the need to improve the economic viability and the rate of production for machined goods. Some of the earliest works in these areas were by empirical equations of Taylor for economic viability and Martellotti in 1941 for understanding the machining process [5]. Martellotti developed an equation of the chip thickness:

$$t_c = R + F_t \sin(\alpha_2) - \sqrt{R^2 - F_t^2 \cos(\alpha_2)^2} \quad (1)$$

where R is the radius of the cutting tool, F_t is the feed per tooth also known as the chip load on each tooth of the cutting tool, and α_2 is the angle of tooth two with respect to a line perpendicular to the feed direction and emanating from the center of the tool [5]. He noted that for spindle speeds that were significantly higher than the feed rate of the tool, this relationship could be reduced to the production of the chip load and the sine of the angle. This common approximation of the instantaneous chip thickness has become the bases of many analytical models that base their analysis on uncut chip thickness. Other analytical models, such as those based on the area of the shear plane or those that attempt to account for the non-linear mechanisms underlying the process do not rely on this approximation of instantaneous chip thickness, but none can consistently and accurately predict the conditions of common real-world machining operations [6].

Models that focus on the phenomena in the thin area surrounding the shear plane have been thoroughly studied, most famously by Eugene Merchant. Milling operations are generally done with cutting tools that contain more than one flute or cutting edge; however, modelling the effects of multiple cutting teeth simultaneously removing material from a part can be quite complex. Instead it is common for models to simplify the problem to the case of a single cutting edge where the orientation of the edge is perpendicular to the direction of travel. This simplified model is referred to as orthogonal cutting. While some of the most famous models, such as Merchant's model [7], use this orthogonal scenario to analyse the cutting process, it is not strictly true for multi-flute cutters. As Ernst first noted in 1938, there are three primary groups that chips can fall into: discontinuous or segmented

chips, continuous chips without a built up edge, and continuous chips with a built up edge [8]. It should be noted that these observations of chip behaviour for both type 1 and 3 chips appear outside of the bounds of the assumptions commonly used in the models discussed, however, for large depth of cut relative to the tooth's local motion these analyses have offered fair approximations [9]. Other analysis such as those conducted by Armarego and Epp in 1969 for straight fluted peripheral milling were able to overcome some of the challenges [9]. Other shear zone models, such as the one developed by Oxley et. al in 1963 attempt to use the physical mechanisms of the cutting process to be able to scale up and predict the cutting forces, but had little success in terms of precision [10].

Due to the shortcomings of the shear plane and shear zone models, research in the field turned towards experimental methods for empirically derived cutting force coefficients, but the values obtained are only strictly valid for the specific parameters of the workpiece and tool for the scenario they were derived from. Thus, these values have little use in terms of developing a robust model that can be used to improve industry. Therefore, it is a third method that won out and proved the most effective and consequently the most used. The mechanistic methods developed early on by Koenigsberger and Sabberwal [11]. These models are based on the original assumptions of the oblique cutting model shown in Figure 1 below, but are commonly based on chip load rather than the intimate analysis of the forces at the level of an oblique model and then scaling up to the helical cutter [12].

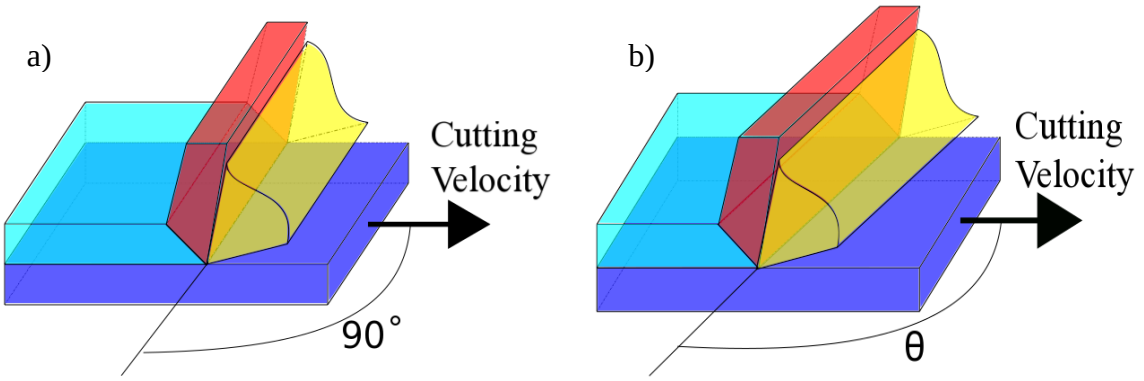


Figure 1. a) Illustration of the orthogonal cutting model. b) Illustration of the oblique cutting model.

Further assumptions made in both the orthogonal and the oblique cutting models are summarized in [7] and include: a perfectly sharp cutting edge on the tool, uniform distribution of the stress in a single thin shear plane that exists ahead of the cutting tool, and no built-up-edge formation. Cutting tools generally have inclined or helical flutes that make assuming orthogonality technically incorrect. Therefore, the case of a single flute can be better represented with an oblique cutting model that allows for the orientation of the cutting edge to be inclined with respect to the cutting velocity.

In mechanistic models the calculation of chip thickness is often approximated using Martellotti's original approximation of chip thickness. His analysis is based on the motion of the cutting flutes as they travel along the trochoidal path [13]. The form of the mechanistic model has been adapted and improved over the years, but one common factor is that the cross-section of the chip area is calculated by from the product of the instantaneous chip thickness and the length of the discrete disk of the cutting tool. This is due to the calculation of the instantaneous chip thickness at discrete points along the length of the cutting tool, thus the distance between the points is used to estimate the integration of the data down the length of the disk elements [14]. Improvements on the mechanistic

model suggested by Fu et. al and Sutherland et. al in [15, 16] respectively. Sutherland's work took the field even further by offering an early look into predicting the deflection of the cutting tool due to the forces of the milling operation. Armarego proposed the subdivision of the effects of the ploughing and shearing components of the cutting force into separate expressions in his work in 1985 with his linear edge force model [17]. Recent improvements in identifying the cutting force coefficients used have been made by Altintas [18] and Azeem et. al [19]. New efforts are being made into more accurately predicting the cutting forces with cutting force coefficients that are distributed along the depth of cut of the tools axial immersion [20].

The limitations of the basic mechanistic model are apparent from the fact that the unique models have been used for every tool-workpiece engagement scenario and variation of the cutting tool geometry. In this way the standard model for the prediction of the interaction between the cutting tool and the workpiece with Martellotti's the sinusoidal approximation is insufficient for practical machining operations. Thus, just as Sutherland expanded the model to the realm of tool deflections and their effect on the machining path; additional expansions allowing for arbitrary geometry and workpiece engagement were developed [21].

1.2.2 Modelling of the Cutting Tool Geometry

Childs helped develop the enveloped cutter geometry commonly used in modern CAM software [22]. Geometric models of cutting tool geometry and how they interact with the workpiece geometry were developed in [23, 24, 25, 26, 27]. The major drawback of these models that represent both the tool and workpiece with surface geometry, is that

the cutter engagement regions require extremely complex boundary equations for estimating the regions of contact from the projection of the workpiece surface onto the surface of the cutting tool. In fact, once the circumstances of the cutter engagement expand beyond even some of the simplest prismatic cuts against rectangular workpiece volumes, the equations of the boundary must be solved with numerical solvers [28].

Other formulations of workpiece and end mill geometry have been proposed such as the z-map technique employed in Ref. [29]. This 2.5D method works well for medium to low complex part geometries, but since the model cannot handle overhangs or similar geometries that occur in the z-axis direction. As such, this model cannot be extended to handle 5-axis machining. The formulation of the tool geometry proposed by Engin and Altintas in Ref. [27] is important in that it is one of the few models that attempts to directly account for the helical flutes of the tool in the definition of the tool geometry. The incorporation of these models with the development of computer aided modelling software was carried out by again Altintas in 1991 [24], and separately by Armarego and Deshpande [30]. Improvements in these computational methods were made by El-Mounayri et al. in 1997 with the incorporation of generalized geometric modelling of the workpiece with the tool [31].

1.3 Drawbacks of Traditional Models and Advantages of Modern Computing Hardware

1.3.1 Drawbacks of Traditional Models

One of the biggest shortcomings of the traditional formulations of the mechanistic model is that it has to be specialized for every new tool geometry and unique workpiece

engagement circumstance. Due to this, the model works only for the cases it has been validated against and was only able to provide limited insight to how to improve general machining practices. This lack of robustness was partially solved by the advent of using envelope surfaces of the tools in the model to allow for the local engagement of the tool with the workpiece to be calculated from the projection of the workpiece geometry onto the tool surface. However, even with this technique, instantaneous local engagement cannot be predicted nor can chip thickness. Therefore, standard models for the thickness of the chip had to still be employed. The geometric tool model, developed in Ref. [28], does take steps to incorporate the outermost helix of the cutting flute in the cutter engagement region mapping, but it still suffers from same need for a numerical solver when the equations of the engagement region's bounds become too complex for analytical solutions. As such, it is clear that surface models do not present a computationally efficient method of calculating the interaction between the milling tool and the workpiece material.

Another drawback of traditional machining models is they become more complex with the increasing complexity of the tool and workpiece contact or simply with the increase in the complexity of the tool path itself. This increased complexity means the boundary equations become more difficult and ultimately the computational load of the model increases. Ultimately, as the complexity of the interaction in surface based models increases the need for numerical solvers and computational power grows. This is unfortunate because it places these methods out of line with the demands of an industry that is implementing CNC machining at greater rates.

1.3.2 Development of Alternative Models

One growing alternative to the traditional analytical models, the surface based representations, and even the 2.5D z-map implementations, is the use of voxels in the simulation of the machining process. The emergence of automated path planning software for 5-axis CAM solutions, such as SculptPrint [32], is an indicator of the advantages that can be obtained by the choice of a different data structure from the traditional surface representation models. Not only do voxels allow for true solid modelling of the workpiece volume, but they also increase the ease and effectiveness of creating offsets of the part surfaces for the development of tool paths and analyses. Other works in the realm of voxel based machining models include Refs. [33, 34, 35]. The approach used in the present work uses a voxelized workpiece to represent the workpiece material with a floating point point-cloud of the tool.

1.3.3 Leveraging Modern Computational Advances

As the advances in modern computing continue to grow demand for completely digital manufacturing processes is becoming a common demand of industries. Thus, leveraging the recent leaps and bounds in computer hardware for the calculation of machining simulations seats well with current trends in industry. Moreover, due to the improvements in specialized computing hardware for graphics processing, the goal of simulating the interaction between a workpiece and a given tool is attainable in a discrete fashion. Leveraging the parallel processing power of multi-core CPU and general-purpose graphics processing units (GPGPU), voxel based simulation of the interaction of a discrete workpiece and an arbitrary tool is possible. The use of voxel modelling for path planning

in automated 5-axis machine tool paths has already been established [36, 37]. Fitting with the trend in digital manufacturing, a model of tool-workpiece interaction based on a voxel framework can co-opt robust computer graphics processes for accelerating computation in this significant area of opportunity.

1.4 Organization of Thesis Manuscript

The purpose of this research is to describe a model for simulating the interaction between a cutting tool and workpiece in the machining process. The objective of the model is to predict cutting forces and workpiece geometry resulting from each tool pass. As such, the following manuscript specifies the particulars of this model through descriptions of its four component models. In this first chapter the background and motivation are discussed, prior research is reviewed, and the problem is defined. The second chapter describes the development of each component model. Tessellation of space into discrete voxels, representation of the cutting tool, calculation of volume removal with chip thickness, and a variation of a mechanistic cutting force model are posed. The model is validated against analytical and experimental data in the third chapter. Then in the fourth chapter the model is expanded to simulate consecutive machining passes. Conclusions are drawn and future work is speculated in the last chapter.

CHAPTER 2. METHODOLOGY

2.1 Voxel Model of the Workpiece

A voxel is a cubic element that is used to tessellate space into uniform segments of volume. Voxels can be used to store information about the state of the space they represent, as well as their relationship to surrounding voxels. As a cubic element every voxel represents a finite volume of material with specific and equal side lengths. Voxels can be easily stacked and grouped in 3-dimensions, face-to-face, in aligned arrays, allowing them to be readily interpreted by computers. As shown in Figure 2, visualization of voxels as simple cubic boxes in space is a simple function of plotting the this binary array.

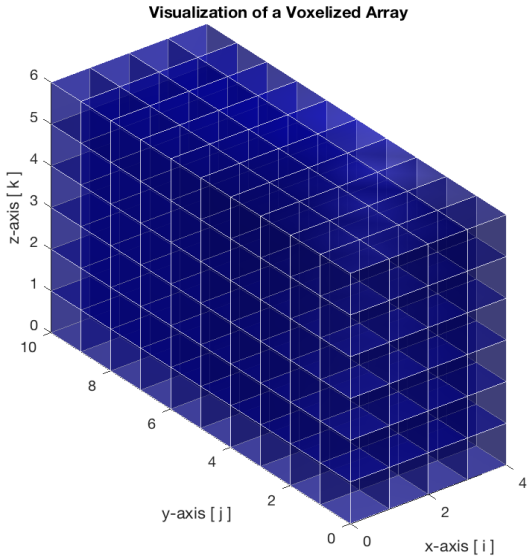


Figure 2. Visualization of a voxelized array.

This allows voxels to naturally function as the simplest tessellation of 3D space. Given the cubic nature of voxels; voxel adjacency can be defined by shared faces, edges, and corners. The cumulative effect is that each voxel can have 26 neighboring voxels each with unique information about the space that they represent.

One of the simplest ways voxels can be used to describe the space they represent is to acknowledge the solidity or fill of their space. The simplest and most computationally efficient representation of this space is through the use of binary information. A voxel with the logical value of ‘true’, or in binary 1, can be thought of as a filled voxel that represents solid or space. This can be understood as a voxel with a value of 1 representing a portion of an object, whereas, a voxel with a value of 0 can be thought of as representing unfilled or empty space. As this simplest of descriptions of their space, voxels can be efficiently used to describe not only the dimensions, and geometry of a part. They can also simultaneously represent the parts total volume through the sum of their discrete volumes. In so much that volume can be defined, traits such as mass density and the part centroid can all be simply derived from the voxelized space with extreme computational efficiency.

Unlike points in space voxels are not infinitely small, thus the representation of where the boundaries of parts are located can be difficult. This can be visualized in attempting to represent an infinitely thin plane with an array of voxels. If these voxels are located with one face coplanar to the plane, then the variation is a full width of a voxel. If, however, the voxels are located with their center points lying on the plane, then the largest variation is only half of a voxel width. These examples are shown below in Figure 3.

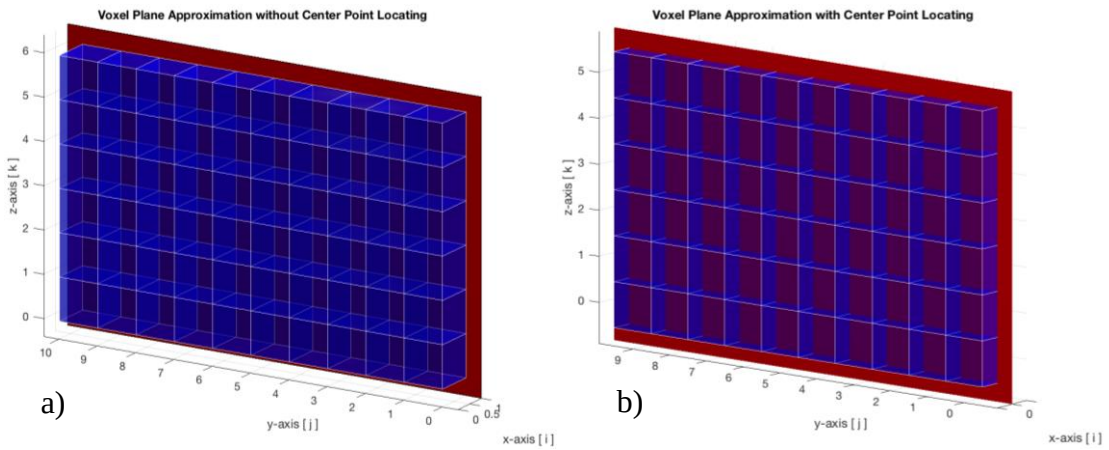


Figure 3. a) Approximation of a thin plane with voxels located their vertices. b) Approximation of a thin plane with voxels located by their centers.

While these examples are ideal and the voxels in question lie perfectly aligned and coplanar versus centered to the plane respectively, the importance of specifying where each voxel's local origin is located is not lost. Additionally, locating voxels by their center point avoids inherent directional bias in representing part geometry, calculating chip thickness, and subsequently cutting forces. Therefore, the voxels in current model are located by their body center point or centroid. The representation of the voxelized blanks with 3D arrays of binary data allows for the physical spacing of the voxels to be stored as a single number that represents the isotropic voxel length, w. A voxel with a width, w, of 0.5 mm is shown below in

Figure 4.

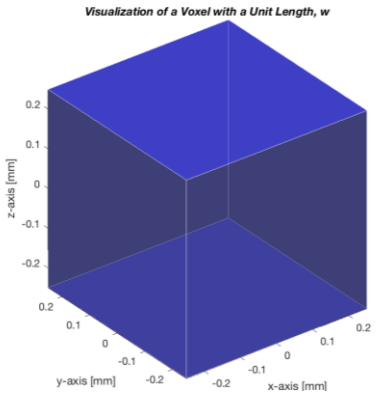


Figure 4. Voxel of isotropic length, *w*.

This makes recovering the location within the part straightforward, in that the location is defined by the 3 indices of the particular voxel multiplied by the predefined voxel length. Furthermore, finding the global location of a voxel is as simple as adding the offset of the workpiece's lowest 3 voxel indices. To illustrate the locating of the voxelized array in

physical space an example of a $5\ mm\ \times 10\ mm\ \times 7\ mm$ voxelized blank is shown below offset from the origin of the global coordinate system by $0.5\ mm\ \times -1\ mm\ \times 1\ mm$ in the x-, y-, and z-directions respectively.

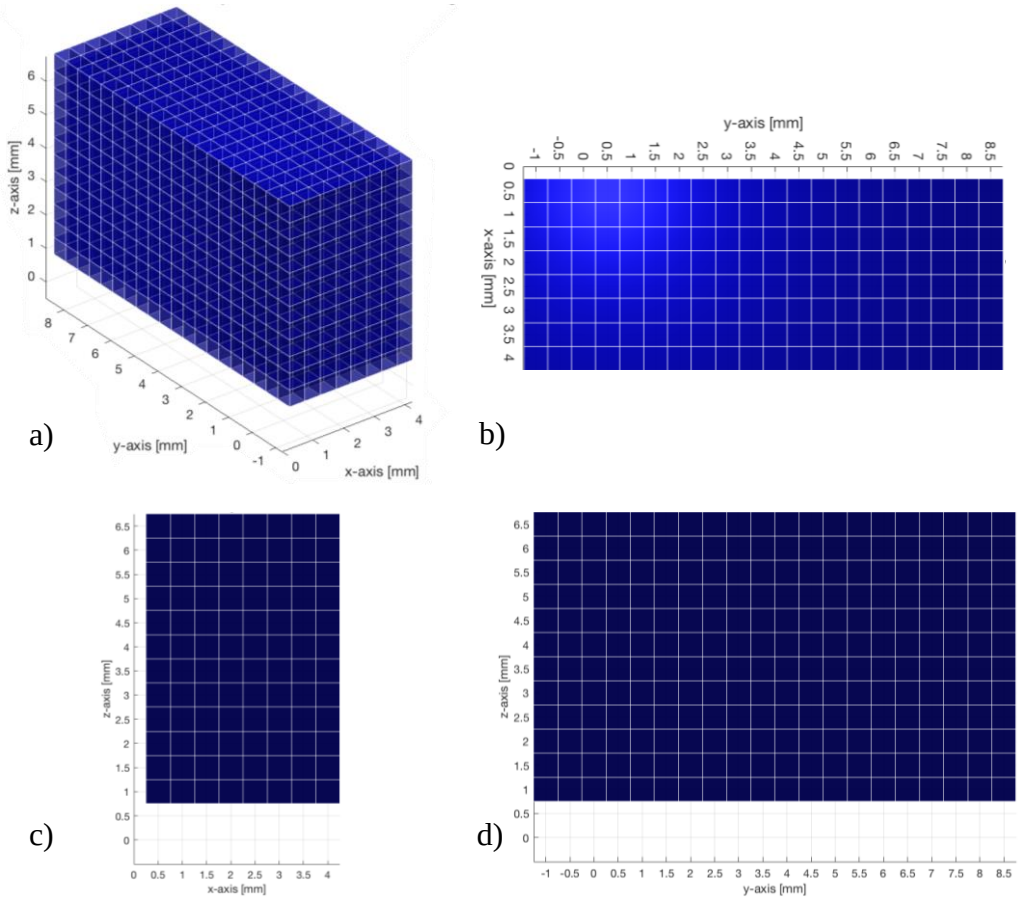


Figure 5. a) Illustration of the example offset of the voxelized workpiece from the global coordinate system. b) XY plane of the example. c) XZ plane of the example. d) YZ plane of the example.

While voxels can be used to represent these properties, their drawback is that they cannot be used to represent a continuum of material. Moreover, the representation of extremely fine details about an object require even finer voxel sizes. Therefore, representing small details about parts will require that the data structure of voxels be as light as possible to allow for efficient and practical applications. In this model adjacent voxels represented by 3D arrays of binary numbers are used to represent rectangular part

blanks before the milling process and the sufficiently accurate representations of their geometry after the milling process. More complex starting geometries can be used through the simple application of constructing the object from uniformly spaced image slices of the object in question are reconstructed into a 3D array. This requires that the image slices be spaced at exactly the same width as desired for each pixel of the image slices. Through this reconstruction, complex workpieces can be made that represent the pre-machining or in-process machining geometry to be used for the milling operation.

2.2 Floating Point Model of the Cutting Tool

Given the large variety of cutting tool geometries used in machining, it is important that any sufficiently adaptive simulation be able to represent the range of cutting tools commonly used in machining operations. For the purpose of practicality this model will be limited to the representation of unibody helical mills only instead of considering the wide levels of variation and specific cutter geometries used with indexable mills. That being said the model is still capable of fully representing simple cylindrical mills, ball mills, bull-nosed mills, tipped mills, and tapered mills, as well as any valid combination of these geometries with continuous curvature.

Representation of cutter geometry is something of importance at all levels of modern machining operations, however the level of accuracy required does differ across representations. For modern computer aided machining (CAM) programs, tools are represented by a simplification of their outermost geometry [6]. This geometry is used to identify the radially symmetric envelope of the cutting tool so that the intersection of this boundary and the part surfaces can be calculated and graphically displayed. While envelope

geometry does offer a simplified representation of the cutter geometry that is computationally efficient, it lacks the necessary geometric description needed to analyze the cutting mechanics.

Cutting mechanics can be described by a single cutting edge orthogonal to the velocity between the tool and the workpiece. If the edge is at an angle with respect to the cutting velocity, then a more geometrically complex model is needed. These two configurations are referred to as orthogonal and oblique cutting respectively. While the majority of cutting operations are the result of two or more cutting edges at inclines to their local cutting direction, these configurations are normally used as the basis of higher level models.

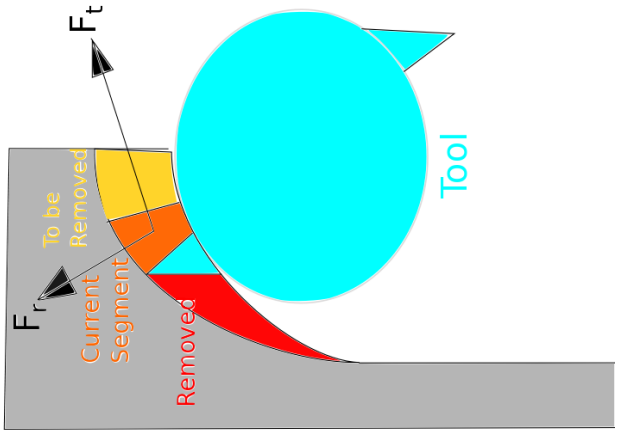


Figure 6.Illustration of how a tooth of a cutting tool makes contact with the workpiece material.

Calculation of the cutting tool and workpiece interactions, requires accurate representation of tool geometry. As one of the primary assumptions of orthogonal and oblique cutting models, only the outermost edge of the helical flute is assumed to be in direct contact with the machined surface of the workpiece. Accordingly, by approximating

the cutting edge of the tool to be infinitely sharp, the primary cutting surface of the tool can be modelled with the rake face leading to this cutting edge. As can be seen in Figure 6, only the outermost edge of the rake face is in contact with the workpiece surface shown in dark blue. Figure 7 illustrates this interaction of the cutting flute in contact with the workpiece and deformed chip in the cutting process. Models for the combination of the cutter envelope and outermost edge of the helical flutes have been developed [27], but these fail to account for the remaining rake face geometry.

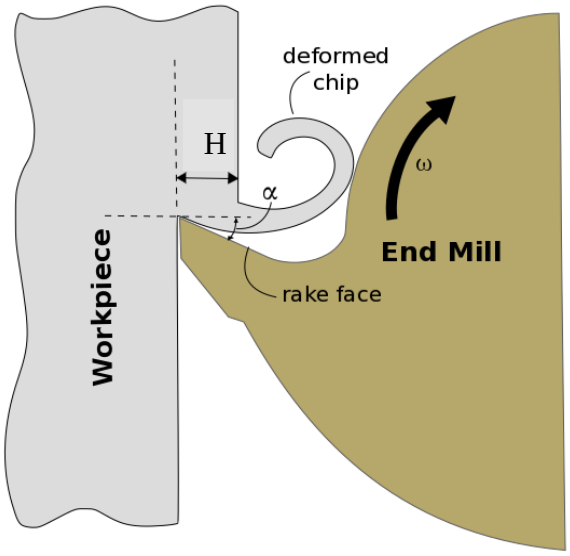


Figure 7. Illustration of a helical mill cutting a workpiece in 2D.

As can be seen in Figure 6 and Figure 7, limiting the modeling of the tool geometry to only the outermost edge of the helical flute along the cutter surface envelope is not a sufficient representation of a tool that is capable of removing workpiece material; information about the rake face is also necessary. The tool model developed for this research shares some of the nomenclature employed by J. J. Childs in Ref. [22]. Starting at the tool tip specific points along the envelope of the cutter are calculated and used to anchor the helix of the tool at specific radii and distances along the tool axis. In the model

presented the user defines at most five geometric parameters, D, R, α, β, & h that allow for the complete description of the tool envelope as seen in Figure 8. This reduced number of geometric parameters from the seven that are required in the generalized model, allows for a more intuitive tool generation for the user by imposing curvature continuity amongst the sections of the milling tool when tools with cutter radii are used. To simplify the prior knowledge required by the user the present model does not allow for the generation of form cutters, unlike the generalized model presented by Engin and Altintas [27]. As seen below in Figure 8, the five parameters can describe a general endmill of arbitrary geometry so long as the radii R is tangent with both line OM and line NS. In cases where the radii equal zero, this tangency is carried out by M and N residing on the same point.

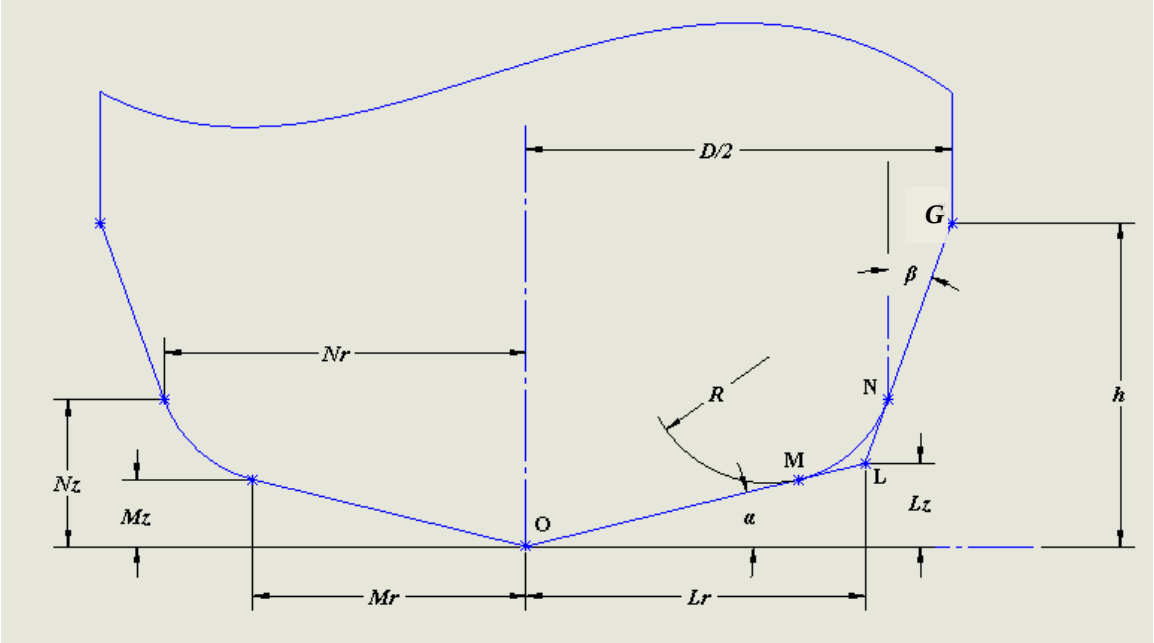


Figure 8. Milling cutter envelope geometry.

Calculating the center of curvature for the radii portion of the tool, shown above in Figure 8 as the region of the cutter envelope that is defined by R, is handled by sorting the

tool type from the user inputs. The definition of the cutter envelope for standard end mill geometries is described by equations (2)-(31) below.

For all endmill geometries:

$$R_r = M_r - R * \sin(\alpha) \tag{2}$$

$$R_z = M_z + R * \cos(\alpha) \tag{3}$$

$$G_r = \frac{D}{2} \tag{4}$$

$$G_z = h \tag{5}$$

For a general endmill:

$$L_r = \frac{h - \frac{D}{2} * \cot(\beta)}{\tan(\alpha) - \cot(\beta)} \tag{6}$$

$$L_z = L_r * \tan(\alpha) \tag{7}$$

$$M_r = L_r - R * \tan\left(\frac{\frac{\pi}{2} - \alpha - \beta}{2}\right) * \cos(\alpha) \tag{8}$$

$$M_z = L_z - R * \tan\left(\frac{\frac{\pi}{2} - \alpha - \beta}{2}\right) * \sin(\alpha) \tag{9}$$

$$N_r = L_r + R * \tan\left(\frac{\frac{\pi}{2} - \alpha - \beta}{2}\right) * \sin(\beta) \tag{10}$$

$$N_z = L_z + R * \tan\left(\frac{\frac{\pi}{2} - \alpha - \beta}{2}\right) * cos(\beta) \quad (11)$$

For a tipped bull-nose endmill:

$$M_r = \frac{D}{2} - R * \left(1 - \cos\left(\frac{\pi}{2} - \alpha\right)\right) \quad (12)$$

$$M_z = M_r * \tan(\alpha) \quad (13)$$

$$N_r = \frac{D}{2} \quad (14)$$

$$N_z = M_z + R * \sin\left(\frac{\pi}{2} - \alpha\right) \quad (15)$$

For a tapered ball or bullnose endmill:

$$N_z = R - R * \cos\left(\frac{\pi}{2} - \beta\right) \quad (16)$$

$$N_r = \frac{D}{2} - (h - N_z) * \tan(\beta) \quad (17)$$

$$M_r = N_r - R * \sin(\frac{\pi}{2} - \beta) \quad (18)$$

$$M_z = 0 \quad (19)$$

For a ball or bullnose endmill:

$$M_r = \frac{D}{2} - R \quad (20)$$

$$M_z = 0 \tag{21}$$

$$N_r = \frac{D}{2} \tag{22}$$

$$N_z = R \tag{23}$$

For a cone endmill:

$$M_r = N_r = \frac{h - \frac{D}{2} * \cot(\beta)}{\tan(\alpha) - \cot(\beta)} \tag{24}$$

$$M_z = N_z = M_r * \tan(\alpha) \tag{25}$$

For a conically tipped endmill:

$$M_r = N_r = \frac{D}{2} \tag{26}$$

$$M_z = N_z = \frac{D}{2} * \tan(\alpha) \tag{27}$$

For a conically tapered endmill:

$$M_r = N_r = \frac{h - \frac{D}{2} * \cot(\beta)}{\tan(\alpha) - \cot(\beta)} \tag{28}$$

$$M_z = N_z = 0 \tag{29}$$

For a standard cylindrical endmill:

$$M_r = N_r = \frac{D}{2} \tag{30}$$

$$M_z = N_z = 0 \tag{31}$$

For the user specified number of flutes and helix angle, the helical outermost edge of the tool is generated by first dividing each section of the helix into an appropriate number of discrete points with arc length ds. The estimated arc lengths of the sections OM, MN, NS normalized by the total arc length of the cutter envelope is used to distribute the total number of discrete points along the cutter's length to each section. The radius from the tool's axis is calculated linearly for OM and NS, and as a function of the angle φ_R for MN.

$$\varphi_0 = \sin^{-1}\frac{M_r - R_r}{R} \tag{32}$$

$$\varphi_{R_\zeta} = \frac{\sum_{\zeta=M}^{N} ds(\zeta)}{R} \tag{33}$$

where φ_R is the angle along MN of the current discrete point.

$$\begin{aligned} r(O{:}M) &= \left(ds(\zeta) + \sum_{idx=0}^{\zeta} ds(idx) \right) * \cos(\alpha) \\ r(M{:}N) &= R_r + R * \sin\left(\varphi_0 + \varphi_{R_\zeta}\right) \\ r(N{:}S) &= N_r + \left(ds(\zeta) + \sum_{idx=0}^{\zeta} ds(idx) \right) * \sin(\beta) \end{aligned} \tag{34}$$

The location along the tool's center axis is calculated by:

$$z(O:M) = \left(ds(\zeta) + \sum_{idx=0}^{\zeta} ds(idx) \right) * \sin(\alpha)$$

$$z(M:N) = R_z - R * \sin\left(\frac{\pi}{2} - \varphi_0 - \varphi_{R_\zeta}\right) \tag{35}$$

$$z(N:S) = N_z + \left(ds(\zeta) + \sum_{idx=0}^{\zeta} ds(idx) \right) * \cos(\beta)$$

The angle between the discrete point and the axis of the tool is constant for both OM and NS, but is a function of the normalized difference between the position from the tool's axis and the location of the radii center point relative to the tool's axis.

$$\kappa(O:M) = \alpha$$

$$\kappa(M:N) = \sin^{-1}\left(\frac{r(M:N) - R_r}{R}\right) \tag{36}$$

$$\kappa(N:S) = \frac{\pi}{2} - \beta$$

This angle plays an important role in determining the cutting forces in the mechanistic model discussed later in this work. For a helix angle, ψ, the angular displacements (γ) around the central axis of the cutting tool are calculated by:

$$\gamma(O:M) = \frac{z(O:M)}{r(O:M)} * \tan(\psi) \tag{37}$$

$$\gamma(N{:}S) = \left(\frac{z(N{:}S) - N_z}{r(N{:}S)} + \frac{(R + N_Z) - R_z}{N_r} - \frac{R - R_z}{M_r}\right) * \tan(\psi)$$

where if $M_r = 0$, the terms containing M_r in the denominator equal zero. Finally, the angle in terms of radians of each discrete point wrapped around the tool for the first flute of the tool is calculated as the tool's spindle speed (ω) in rpm multiplied by the total time of the simulation (τ_{total}) in seconds.

$$\Phi_{referenceflute} = mod\left(\frac{\omega}{60seconds} * 2\pi * \tau_{total} - \gamma, 2\pi\right) \tag{38}$$

where the mod ensures that the tool angles are between 0 and 2π. To simplify and group the following information about the cutting tool, the 'structure' convention will be used for the tool points from this point forward. Movement ($P.tip$) along any arbitrary direction can be defined by the product of the feedrate in distance traveled per minute with a normalized vector of the feed direction ($\vec{e}_{feedrate}$) and the total time of the motion step (τ_{total}) as seen in equation (39):

$$P.tip = \frac{Feedrate}{60seconds} * \vec{e}_{feedrate} * \tau_{total} + C \tag{39}$$

where C is the user-defined starting point. The above equation just defines the location of the tool tip over the time τ_{total}. To apply the motion to the rest of the tool, the angular location of each flute with respect to the tool axis must be defined. As such, with the user defined number of cutting flutes N_{flutes}, the angular coordinates of each flute with respect to the tool's axis are defined by:

$$\Phi_{flutes} = mod\left(\Phi_{referenceflute} + (\Gamma - 1) * \frac{2\pi}{N_{flutes}}, 2\pi\right), \quad for\ \Gamma = 1\ to\ N_{flutes} \tag{40}$$

where again the mod limits the realm of possible angles between 0 and 2π. With the necessary position of all of the cutting flutes defined, the physical location of the discrete points for each flute can be defined as:

$$P.coord = r * \sin(\Phi_{flutes}) * \overrightarrow{e_\imath} + \mathrm{r} * \cos(\Phi_{flutes}) * \overrightarrow{e_\jmath} + z * \overrightarrow{e_k} + P.tip \tag{41}$$

where $\overrightarrow{e_\imath}$, $\overrightarrow{e_\jmath}$, $\overrightarrow{e_k}$ are the unit vectors for the x, y, and z directions respectively. By adding the movement of the tool's tip, $P.tip$, to equation (41), the location and translation of the tool points in time is defined. The length of the rake lines (L_{rake}) can be defined by the user, but normally set to be a function of the diameter of the cutting tool. With a user defined rake angle (α_{rake}), the rake line can be created by specifying a secondary point interior of the tool. The line between the interior point and its corresponding outermost flute point is the rake line used to represent the local rake face of the tool.

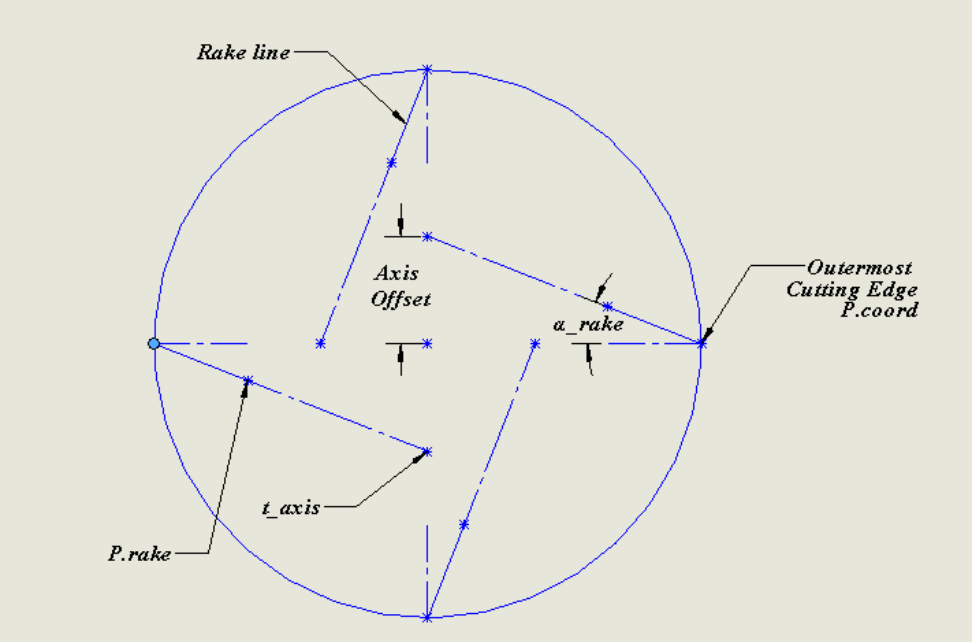


Figure 9. Illustration in 2D of the geometry of the rake line calculation from the offset of the tool axis.

As can be seen in Figure 9, the rake line is made by offsetting the tool's axis, calculating the vector between the outermost cutting edge of the tool and the offset axis, and finally calculating the vector point for the rake as the sum of the outermost cutting edge with the product of the rake face length (L_{rake}) and the normalized vector between the offset axis and the outermost cutting edge as shown in equation (42).

$$\begin{aligned} Offset\ Axis \ &\stackrel{\text{def}}{=} t_{axis} = P.tip - (P.tip - P.coord) * \tan(\alpha_{rake}) \\ P.rake &= P.coord + L_{rake} * \frac{t_{axis} - P.coord}{\|t_{axis} - P.coord\|} \end{aligned} \tag{42}$$

The mathematical description of the milling tool and its motion above is sufficient for 3-axis milling operations; however, in modern CNC machining operations, multiple rotational axes are often used. While most multi-orientation machine tools are limited to 5-axes, since there is not a consensus amongst machine tool manufacturers as to which of the

primary axes should be allowed to rotate, this model will incorporate three additional rotational axes for a total of six degrees of freedom in the specification of tool movement relative to the workpiece, or simply the ability to model a 6-axis machine tool.

Let angles μ, η, & λ represent the rotations about the A-axis, B-axis, and C-axis, respectively. With the A-axis, B-axis, and C-axis defined as the rotations about the x-axis, y-axis, and z-axis of the machine coordinate system respectively, the combined rotation matrix for this motion at time τ_ξ can be defined as follows:

$$c_{\sim_\xi} = \cos(\sim_\xi) \quad \& \quad s_{\sim_\xi} = \sin(\sim_\xi)$$

$$\boldsymbol{R}_\xi = \begin{bmatrix} c_{\eta_\xi} c_{\lambda_\xi} & -c_{\eta_\xi} s_{\lambda_\xi} & s_{\eta_\xi} \\ c_{\mu_\xi} s_{\lambda_\xi} + s_{\mu_\xi} s_{\eta_\xi} c_{\lambda_\xi} & c_{\mu_\xi} c_{\lambda_\xi} - s_{\mu_\xi} s_{\eta_\xi} s_{\lambda_\xi} & -s_{\mu_\xi} c_{\eta_\xi} \\ s_{\mu_\xi} s_{\lambda_\xi} - c_{\mu_\xi} s_{\eta_\xi} c_{\lambda_\xi} & s_{\mu_\xi} c_{\lambda_\xi} + c_{\mu_\xi} s_{\eta_\xi} s_{\lambda_\xi} & c_{\mu_\xi} c_{\eta_\xi} \end{bmatrix} \tag{43}$$

Then to calculate tool orientation at time ξ the following equation is used:

$$\begin{aligned} P.coord_\xi &= \boldsymbol{R}_\xi * \left(P.coord_\xi - P.tip_\xi\right) + P.tip_\xi \\ P.rake_\xi &= \boldsymbol{R}_\xi * \left(P.rake_\xi - P.tip_\xi\right) + P.tip_\xi \end{aligned} \tag{44}$$

With the complete models of the milling tool and the workpiece defined, the interaction between the workpiece and the cutting tool can be thought of as the deletion of voxels by the rake line of the tool once it passes into their boundary.

2.3 Ray Casting Model of the Interaction

The successful simulation of the machining process is largely dependent on the approach to calculating the interaction between the cutting tool and the workpiece. As

discussed the approach taken in this work is to simulate the physical space and material of the workpiece with a volumetrically digitized framework of binary data; while the tool is represented by a point cloud of the outermost cutting edge points and their corresponding interior points that represent the termination of the rake face just before the gullet of the tool's cutting flute. With a number of discrete time steps the motion of the tool is captured, and as the milling tool is fed forward and rotated about its axis, the rake lines will move into contact with different voxels of the workpiece. Through this superposition of the two component models, the process of removing workpiece material is mimicked by the value of the voxel in contact with the tool being changed from one to zero effectively simulating their removal from the workpiece.

In order to efficiently calculate which voxels are pierced by a rake line of the tool, lightweight and highly optimized algorithms are needed. This is where the field of computer graphics techniques comes into play. Although there are other computer graphics based techniques for calculating intersections, many rely on iteratively refining the local geometry around the intersection to improve the accuracy of the calculation until a point of sufficient accuracy is reached. Since the workpiece is represented down to the desired level of accuracy by voxels to begin with, there is no need to refine the local geometry in real time with the simulation. For the same reason, calculating the intersection using these methods that rely on surface geometry is also inefficient. Instead, a method is needed that can efficiently and accurately calculate which voxels have been pierced by a local rake line. As an advantage of the simplicity in representing the workpiece as a framework of cubic elements and the tool motion as distinct sets of lines at specific locations in time, the calculation of their intersection is also simple.

Ray casting is one of the most common methods used to render images and lighting. Due to the growth in the fields of computer generated images (CGI) and video games in recent decades, ray casting has become a highly developed and optimized technique. Moreover, due to the large recent advances in graphics processing unit (GPUs) hardware that can specifically accelerate 3D kernel computations by parallelizing operations across a high number of cores; uniformly discretized data such as voxelized arrays allow for very efficient computational times [38]. Typical ray casting implementations follow four primary steps. First, a ray of light is projected from a given view orientation. The projected ray is then sampled along its length for objects that may be pierced by the straight line. Next, the shading caused by this intersection is calculated and finally the objects that are pierced as well as their calculated shading are assembled along the ray back to the source to determine what is displayed along that path in the image [39]. In this proposed model the rake lines from the tool are used in place of rays of light, and the application is to determine the interaction between the tool and the workpiece. As such, no rendering steps are needed. Modern ray casting techniques can be sorted into two primary categories: continuous and discrete [40].

Continuous algorithms are more intuitive and lend well to verbal description, because at their root they are based on the parametric representation of a line. As the parametric variable is sampled from zero to one, the points along the line are calculated. Unfortunately, this method is not viable for calculating a line's intersection with discrete cubic elements like voxels, because ensuring that at least one point along the line was calculated inside of each voxel that the line passes through theoretically requires an infinite number of points to be calculated along the line. Another drawback of the continuous

method is that multiple points calculated along the line may lie within the same voxel making their calculation redundant. Instead, this work will apply a discrete ray casting technique in which the pierced voxels are generated directly.

Although there are different methods for calculating the voxel traversal of discrete ray casting algorithms, at their root all the discrete algorithms rely on the indexing of the voxels rather than the occurrence of points along the line within a voxel. Within the discrete regime of ray casting techniques some methods rely on the Bresenham algorithm with checks for the non-Bresenham voxels as needed as shown in Figure 10.

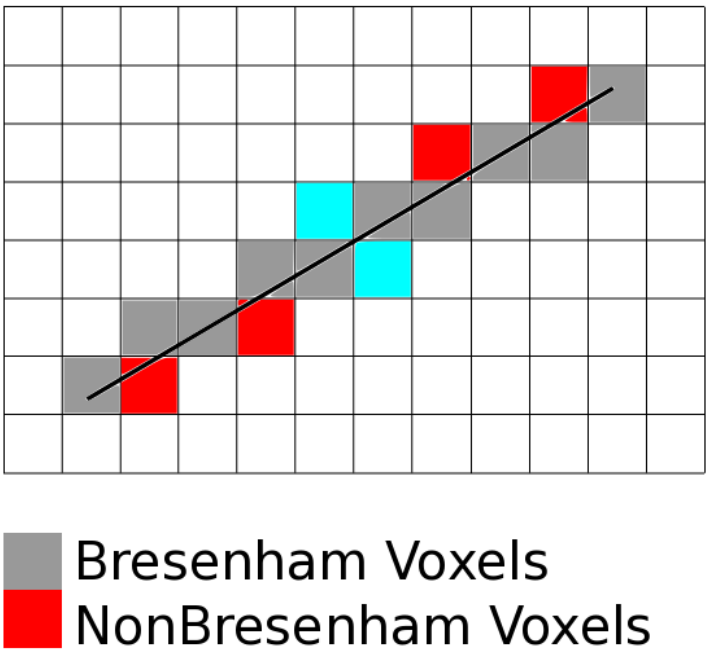


Figure 10. Illustration of the voxels traversed by and the voxels missed by the Bresenham algorithm for voxel traversal.

Figure 10 also shows that when the ray is considered to pass perfectly through the vertex between voxels, as seen by the vertex shared between the cyan voxels, neither cyan voxel is included in the set of voxels pierced by the ray. Unfortunately, for these methods to be accurate, floating point calculations are required to find the non-Bresenham voxels that are

missed. Even if uncommon these occasional floating point calculations slow down the calculation. Instead, methods that calculate the voxel traversal by the facial connectivity between the voxels prove to be faster [40]. This is due to the ability of these algorithms relying on integer rather than floating point calculations.

In the proposed model a modified version of the algorithm developed by John Amanatides and Andrew Woo is used to calculate the voxel traversal by each ray of the tool [41]. The advantages of this model are great in that even when the ray is cast through a very large number of voxels there is no need to recalculate any floating point numbers for the duration of the ray traversal. This allows the model to be approximately 10-25% faster, for small and large numbers of voxels respectively, than Bresenham based algorithms. Moreover, their algorithm proved faster than the original Cleary and Wyvill algorithm for large numbers of voxels [40]. The Amanatides and Woo algorithm is performed in two parts for each ray that is cast: ray initialization and discrete voxel traversal.

The advantage of the ray initialization step is that it finds the first voxel that the ray penetrates, thus limiting the calculation of the ray traversal to only the length of the ray in contact with voxels. The proposed model uses an even more robust initialization step introduced by Brian Smits in 1998 [42] and formalized into implementation by Amy Williams et al. in 2005 [1]. This more robust algorithm allows for the code to check if a ray will ever intersect with a voxel array; and if so, to calculate the floating point value of the parametric variable at the first point along the ray to pierce the voxelized workpiece. This works for voxel volumes that are aligned with the global coordinate system as they are in this proposed method. This step is crucial in the proposed model since the majority

of the tool is generally not in contact with the workpiece at any given instance in time for most milling operations. This allows for the model to avoid unnecessary calculations of traversal along the rake lines not in contact with the tool at each time step of the simulation. The algorithm works by checking the sign of each component of the normalized vector of the ray to differentiate how to assign the min and max value of the parametric variable. Equations (45) and (46) below summarize this functionality:

$$S_{\min}(q) = \begin{cases} \dfrac{Bound_{\min}(q) - P.rake(q)}{\underline{\hat{d}}(q)} & if\ \underline{\hat{d}}(q) \geq 0 \\ \dfrac{Bound_{max}(q) - P.rake(q)}{\underline{\hat{d}}(q)} & else \end{cases} \tag{45}$$

$$S_{max}(q) = \begin{cases} \dfrac{Bound_{\max}(q) - P.rake(q)}{\underline{\hat{d}}(q)} & if\ \underline{\hat{d}}(q) \geq 0 \\ \dfrac{Bound_{min}(q) - P.rake(q)}{\underline{\hat{d}}(q)} & else \end{cases} \tag{46}$$

where $\underline{\hat{d}}(q)$ is the unit vector of the ray's direction. The above equations are set up for the rays defined as emanating from the inner points, $P.rake$, of the tool's rake face. This operation is carried out for all three components, x, y, and z, of the global coordinate system, where these components are represented by the index q. To determine if the ray ever penetrates the workpiece volume, the value $S_{\min}(i)$ of component i is compared with the value $S_{max}(j)$ of component j and $S_{max}(i)$ is compared with $S_{min}(j)$, where $i \neq j$. If the maximum value of the parametric variable in one direction is less than the minimum value of the parametric variable in another direction, the ray does not penetrate the workpiece before missing the boundary. This relationship is shown below in Figure 11.

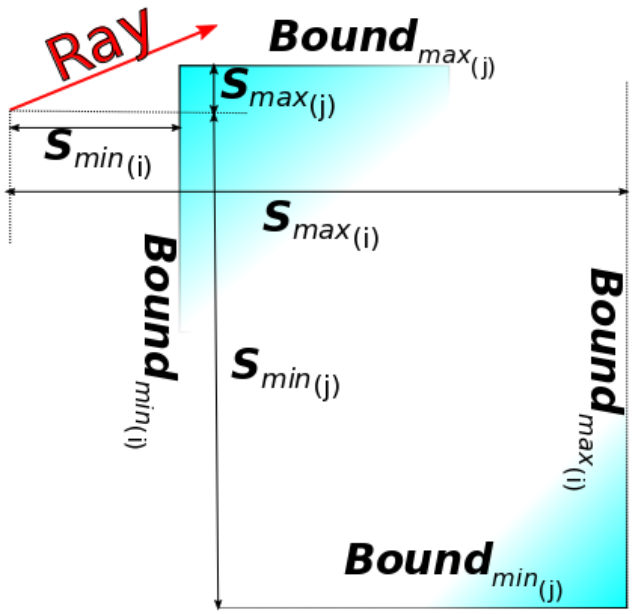


Figure 11. Illustration of the ray box intersection algorithm for determining if the ray penetrates the voxelized volume as described in [1].

If the ray does not miss the voxel volume, the largest S_{min} component is set to the value of the parametric variable, S, at the first voxel pierced by the ray. The first point along the ray to penetrate a voxel is then calculated as:

$$\boldsymbol{p}_{start} = P.rake + S * \underline{\hat{d}} \tag{47}$$

Similarly, finding the index of the first voxel pierced by the ray is as simple as dividing the difference between $\boldsymbol{p}_{start}$ of the ray and $Bound_{min}$ of the workpiece by the voxel size, w, as seen in equation (48) where the values are rounded to the nearest integer.

$$\begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} = \frac{\boldsymbol{p}_{start}\begin{pmatrix} i \\ j \\ k \end{pmatrix} - Bound_{min}\begin{pmatrix} i \\ j \\ k \end{pmatrix}}{w} \tag{48}$$

Then the distance in terms of the parametric variable, S, for the ray to move by the width of a voxel is calculated for each direction, $i, j, and\ k$. To determine this distance with the least number of floating point calculations, the signs of the components of the normalized direction, $\underline{\hat{d}}$, are applied as the sign of the unit step in each component direction, $i, j, and\ k$. Since the ray can only move one voxel at a time, a unit step for each component $i, j, and\ k$ of $\underline{\hat{d}}$ is used and its sign coordinates which direction the next voxel is located in.

$$\begin{pmatrix} \Delta x_i \\ \Delta y_j \\ \Delta z_k \end{pmatrix} = \begin{cases} 1 & if\ \underline{\hat{d}} \begin{pmatrix} i \\ j \\ k \end{pmatrix} > 0 \\ -1 & else \end{cases} \tag{49}$$

The floating point calculation of the next voxel's location in the global coordinate system, $\boldsymbol{p}_{next}$, is calculated from the ratio of the next voxel's indices and their corresponding total number of voxels plus the value of the $Bound_{min}$ for each component direction. This value is then used to find how far the ray must move in each direction in terms of parametric variable, S, as seen in equation (50).

$$dS \begin{pmatrix} i \\ j \\ k \end{pmatrix} = S_{min} + \frac{\boldsymbol{p}_{next} - \boldsymbol{p}_{start}}{\underline{\hat{d}}} \tag{50}$$

Note that dS is set to infinity for any component of the direction $\underline{\hat{d}}$ that is equal to zero. That is to say the minimum distance in terms of the parametric variable, S, needed to move from one voxel to the next in a direction where the ray does not change is set to infinity so that the ray cannot traverse to a different voxel in that direction. Finally, the width of a

voxel, w, divided by the unit direction of the ray, $\underline{\hat{d}}$, defines a separate amount to update each component of dS with each time the ray traverses a voxel in its direction.

$$\Delta dS = \frac{w}{|\underline{\hat{d}}|} \tag{51}$$

With all of the initial parameters defined the algorithm can now calculate the voxels traversed by the ray. Unlike traditional ray casting algorithms, this ray will not terminate at the end of the voxel volume, but rather at the outermost point of the cutting flute. Therefore, the Amanatides and Woo algorithm had to be further altered to allow the termination of the traversal once the last voxel pierced by each rake line was reached. Thus, conditions for the termination of the voxel traversal are needed. The conditions for termination of the ray are defined as follows:

$$\begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{min} = \frac{\min(P.rake \quad P.coord) - Bound_{min}}{w} \tag{52}$$

$$\begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{max} = \frac{\max(P.rake \quad P.coord) - Bound_{min}}{w} \tag{53}$$

$$condition \begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} \geq \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{min} \quad \& \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} \leq \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{max} \tag{54}$$

The above equations define that the traversal calculation will end when the last voxel pierced by the rake line is reached and that it will not calculate before the starting voxel. The traversal algorithm is a while loop that is valid while within these conditions. The component values of the parametric variable, dS, are used to determine the next voxel that

is traveled to by the ray. The ray travels to the voxel in the axis of the smallest component value of the parametric variable, where the direction of that travel is determined by the sign of the unit step previously discussed. The specific component value is this increased by the corresponding component of the ΔdS variable and the process is repeated until the loop conditions are no longer met.

$$\begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} = \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} + \begin{pmatrix} \Delta x_i \\ \Delta y_j \\ \Delta z_k \end{pmatrix} \begin{cases} dS_i < dS_j \; and \; dS_i < dS_k \\ dS_j < dS_i \; and \; dS_j < dS_k \\ dS_k < dS_i \; and \; dS_k < dS_j \end{cases} \tag{55}$$

$$\begin{pmatrix} dS_i \\ dS_j \\ dS_k \end{pmatrix} = \begin{pmatrix} dS_i \\ dS_j \\ dS_k \end{pmatrix} + \begin{pmatrix} \Delta dS_i \\ \Delta dS_j \\ \Delta dS_k \end{pmatrix} \begin{cases} dS_i < dS_j \; and \; dS_i < dS_k \\ dS_j < dS_i \; and \; dS_j < dS_k \\ dS_k < dS_i \; and \; dS_k < dS_j \end{cases} \tag{56}$$

The last part of the tool and workpiece interaction model is the calculation of chip thickness along each rake line. Since the number and direction of the voxels removed is calculated by the voxel traversal anyway, all the information needed to calculate the chip thickness without an analytical expression is already available. Chip thickness, H, is calculated by summing the number of voxels traversed for each component direction and then taking the Pythagorean distance of each component total multiplied by the voxel size, w. To do this, an additional functionality is added to the traversal while loop, that is to keep a count of the number of times a step has been take for each component direction. The following equation summarizes this calculation.

$$H_{\xi,\Gamma,\zeta} = \sqrt{\left(N_{x_i} * w\right)^2 + \left(N_{y_j} * w\right)^2 + \left(N_{z_k} * w\right)^2} \tag{57}$$

The following pseudo code summarizes this tool workpiece interaction model:

For a step in time (ξ)
  For a specific flute (Γ)
    For a specific rake line (ζ)
      While traversing within the rake line
        Step to next closest voxel along the rake line.
        Increase the count in step direction: $N_{x_i,y_j,z_k} = N_{x_i,y_j,z_k} + 1$.
        Check if still within the rake line.
      End while
      $H_{\xi,\Gamma,\zeta} = \sqrt{\left(N_{x_i} * w\right)^2 + \left(N_{y_j} * w\right)^2 + \left(N_{z_k} * w\right)^2}$
    End for
  End for
End for

2.4 Mechanistic Model of the Cutting Forces

At its root, the cutting of metals is a plastic-flow process, but the modeling of this deformation is incredibly complex. Past research has focused on modelling the deformation zone with both thin-zone and thick-zone models. Merchant's orthogonal cutting thin shear-plane analysis is arguably the most famous analysis in these areas but requires a few fundamental assumptions that are not all strictly true for milling operations. These include that the tool tip is perfectly sharp, no built-up edge occurs, and the stress is uniformly distributed in the shear plane [11]. Merchant's model along with many other analytical models cannot precisely predict the cutting mechanics in practical applications.

The remaining methods are generally broken down by their approach to predicting the cutting forces. The first group continues to rely on the orthogonal and oblique cutting models with thin shear zones to improve derived equations [9], while the latter methods use empirically defined equations that rely on prior experimental results often referred to as mechanistic methods [43]. All of the methods mentioned above use similar assumptions including: the tool tip is perfectly sharp, and the chip only contacts the tool along the rake

face of the tool. The model presented in this paper is based on the mechanistic model with alterations to its formulation based on the discrete nature of the data set.

Similarly, the method used in the present model is based on experimentally defined cutting pressure coefficients and the discrete breakdown of the tool at increments in time and in increments along the tool's axis. However, in the present model the discretization is not only of the tool, but also of the workpiece. Because of this, some sampling and smoothing of the output data is needed to enforce a smooth force trace from the interaction of the milling tool and the workpiece material. As seen in Figure 12, the movement of a rake line through a field of voxels even in 2D results in periods where the rake line may not be in contact with a sufficient number of voxels to represent the true chip thickness of a given contact scenario.

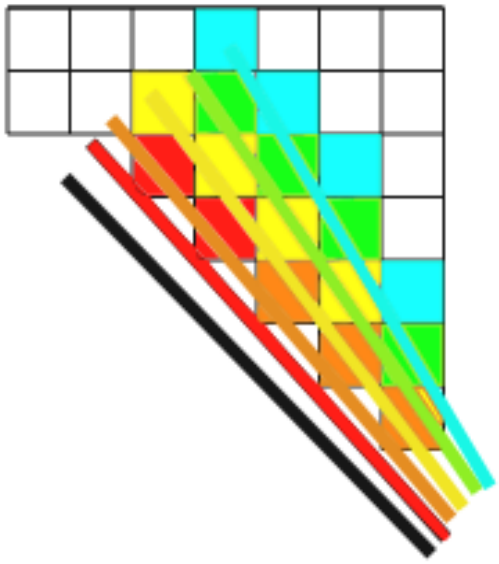


Figure 12. Illustration of how a rake line may move through and contact the voxelized workpiece in 2D.

Due to this inherent mismatch between discrete and analog data, purposeful sampling of the chip thickness data is needed to ensure that the forces are properly calculated. This sampling regime will need to make use of the fact that although the rake line may experience insufficient material contact with the voxelized workpiece for the representation of the true chip thickness, it will never exceed the amount of material contact with the workpiece to non-negligibly overestimate the chip thickness. This allows for the chip thickness data to be sampled within bands of time for the maximum chip thickness, and in doing so, smooth the data to better represent the continuum of the true material at the same time.

Although this implies that the data density of the proposed model is unnecessarily large, the prediction of milling forces is only a small part of the overall objective of this model. Therefore, the reduction of data density required to represent the forces realistically from discretely produced data set allows the model to additionally predict forces from the data made in the interaction of the tool and workpiece. For faster calculations the chip thickness data can also be sampled along the axis of the cutting tool by choosing the maximum chip thickness resulting from a small band of rake lines. It should however be noted that sampling along the rake lines must be done for only small sets of the rake lines, as reducing the data along the axis of the tool does directly affect the resolution of the material interaction. This is, of course, dependent on how small the voxel size is compared to the applied spindle speed and feedrate. Caution should be used when voxel sizes are set to be the maximum width possible to still obtain a certain resolution of the workpiece surface features, as sampling along the axis of the tool will further limit the resolution of the surface features' coherent effects on the milling force prediction.

The proposed alteration of the mechanistic cutting force model is calculated very similarly to the existing model in literature [28] in that the differential cutting forces generated can be calculated by the linear edge force model developed by Armarego in 1969 [9]. The following equation shows how this linear edge force model is implemented in the proposed model:

$$\begin{pmatrix} dF_{r_{\xi,\Gamma,\zeta}} \\ dF_{t_{\xi,\Gamma,\zeta}} \\ dF_{a_{\xi,\Gamma,\zeta}} \end{pmatrix} = \begin{pmatrix} K_{rc} \\ K_{tc} \\ K_{ac} \end{pmatrix} * H_{\xi,\Gamma,\zeta} * w + \begin{pmatrix} K_{re} \\ K_{te} \\ K_{ae} \end{pmatrix} * ds \tag{58}$$

where $dF_{r_{\xi,\Gamma,\zeta}}, dF_{t_{\xi,\Gamma,\zeta}}$, & $dF_{a_{\xi,\Gamma,\zeta}}$ are the differential cutting forces in the radial, tangential, and axial cutting directions respectively. Additionally, ds is defined as the arc length of the flute between rake lines, $\zeta \; and \; \zeta + 1$, along the axis of the milling tool. The cutting force coefficients are divided amongst the cutting $K_{rc}, K_{tc}, \& K_{ac}$ coefficients of the shear zone and the ploughing $K_{re}, K_{te}, \; \& K_{ae}$ coefficients of the frictional contact between the tool and the workpiece [44]. Equation (58) defines the instantaneous elements of the force for every rake line, ζ, on every flute, Γ, at every point in time, ξ individually.

As discussed earlier, the sampling of chip thickness is carried out to smooth the data before computing the force prediction shown in equation (58) above. Associating this sampled chip thickness with the correct angle is accomplished by simultaneously extracting the index of the maximum chip thickness when finding the ideal (maximum) chip thickness within the sampling period. With the chip thickness associated with the correct rotational angel of the simulation, and for more complex geometries the angle off of perpendicular to

the tool axis, the global forces for 3-axis milling operations can be defined by a simple transformation matrix.

$$
\begin{gathered}
c_\kappa = \cos(\kappa_\zeta),\ s_\kappa = \sin(\kappa_\zeta), \\
\mathrm{c}_\Phi = \cos(\Phi_{flutes}),\ \ \&\ \ \mathrm{s}_\Phi = \sin(\Phi_{flutes}) \\
\boldsymbol{R}_{i,j,k} = \begin{bmatrix} -s_\kappa * s_\Phi & -c_\Phi & -c_\kappa * s_\Phi \\ -s_\kappa * c_\Phi & s_\Phi & -c_\kappa * c_\Phi \\ c_\kappa & 0 & -s_\kappa \end{bmatrix}
\end{gathered}
\tag{59}
$$

where κ is the angle from the tool axis, and Φ is the angle of the individual rake line about the tool's axis. It should be noted that since Φ is dependent on the time, ξ, flute, Γ, and rake line, ζ, the rotation matrix, $\boldsymbol{R}_{i,j,k}$, is as well. Multiplying equation (58) by the rotation matrix defined in equation (59), the value of the differential forces in the x_i, y_j, z_k directions of the global coordinate system. To extract these same differential forces for a 3+ axis machine tool the differential forces, $dF_{r_{\xi,\Gamma,\zeta}}, dF_{t_{\xi,\Gamma,\zeta}}$, & $dF_{a_{\xi,\Gamma,\zeta}}$ in the 3+axis tool coordinate system must first be transformed back to the 3-axis tool coordinate system by applying the inverse of the rotation matrix defined in equation (43). The final form of the differential force elements in the workpiece or global coordinate system are:

$$
\begin{pmatrix} dF_{x_\xi} \\ dF_{y_\xi} \\ dF_{z_\xi} \end{pmatrix} = \boldsymbol{R}_{i,j,k} * \begin{pmatrix} dF_{r_{\xi,\Gamma,\zeta}} \\ dF_{t_{\xi,\Gamma,\zeta}} \\ dF_{a_{\xi,\Gamma,\zeta}} \end{pmatrix} \tag{60}
$$

$$
\begin{pmatrix} dF_{x_\xi} \\ dF_{y_\xi} \\ dF_{z_\xi} \end{pmatrix} = \boldsymbol{R}_{i,j,k} * \boldsymbol{R}_\xi * \begin{pmatrix} dF_{r_{\xi,\Gamma,\zeta}} \\ dF_{t_{\xi,\Gamma,\zeta}} \\ dF_{a_{\xi,\Gamma,\zeta}} \end{pmatrix} \tag{61}
$$

for 3-axis and 3+axis machine tool paths respectively. The final product is a series of line running down the rake face of a milling cutter similar to the red lines spanning the rake face of the example cutter shown below in Figure 13. The spacing of these example rake lines does not represent the rake line spacing of the tools generated by the model.

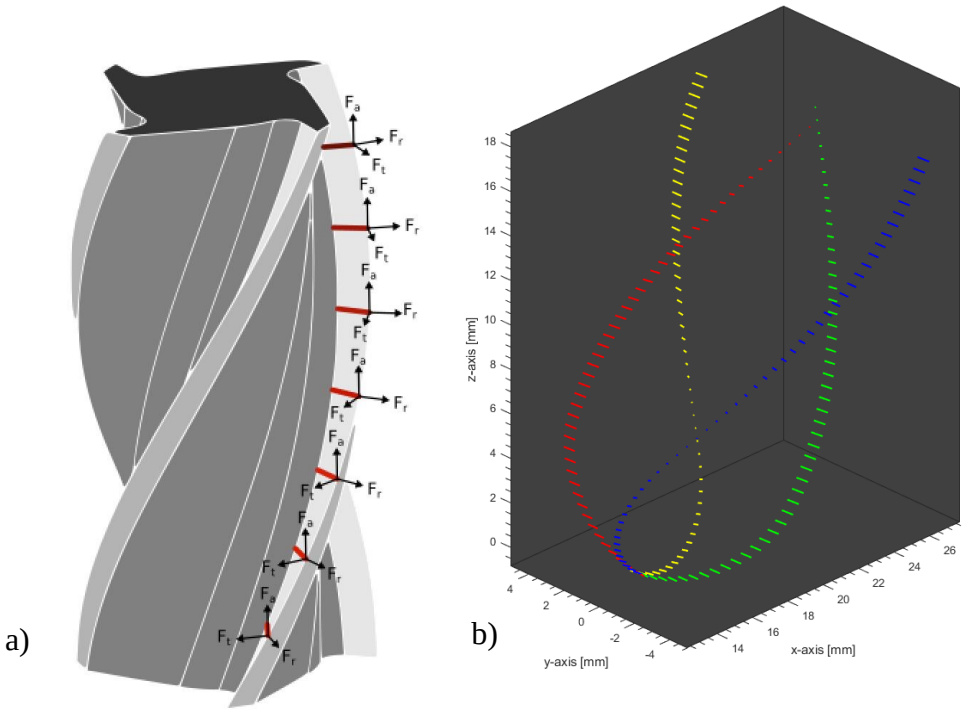


Figure 13. a) Example of a cutter with rake lines, shown in red, spanning the rake face of a flute. The coordinate system tripods for the radial, tangential, and axial direction are shown at the end of each rake line to represent how the forces calculated for the tool are distributed and oriented relative to one another. b) Example of tool generated by the proposed model, where each flute is color coded to show how they wrap around the cutter envelope.

CHAPTER 3. VALIDATION OF THE CUTTING MODEL

3.1 Basic Variation of the parameters of the Component-Models

3.1.1 Variation of Voxel Size

To test the simulation's dependence on voxel size a test scenario was created. The purpose of the test was to measure the effect of voxel size on critical part dimensions and predicted forces. The test consists of a voxelized workpiece blank initially $2\,mm \times 2\,mm\,x\,1\,mm$ in dimension created with three voxel sizes that differ by an order of magnitude: 0.01 mm, 0.05 mm, and 0.l mm. For this test a 15 mm diameter (D) tool with 1 flute at a 30º helix angle (ψ) and a 0º rake angle (α_{rake}) is generated. The tool is run at a chip load (f_t) of 0.3 mm/tooth and a spindle speed (SS) of 1500 rpm. The setup of the tool is 1.8 mm radially immersion (r_{imm}) up-milling with an axial immersion $(a_{imm.})$ of 1 mm simulating high step over peripheral milling. The reason this milling scenario was chosen is to show how accurate the simulation can be with a maximum chip thickness measured over 10 voxels and how inaccurate this can become if the maximum chip thickness is spanned by only a single voxel width. Table 1 summarizes the voxel sizes for each test case and Table 2 summarizes the milling parameters of this milling scenario.

Table 1. Test cases defined by variation of voxel size.

Test Case	1	2	3
Voxel Size (w) [mm]	0.0025	0.01	0.1

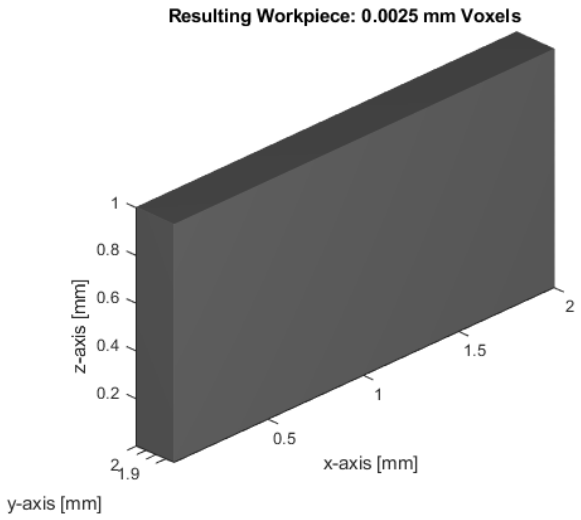
The voxel sizes were chosen so that the workpiece would be $800 \times 800 \times 400\ voxels$, $200 \times 200 \times 100\ voxels$, and $20 \times 20 \times 10\ voxels$ for each test case 1, 2, and 3, respectively. This sets the maximum chip load to be 120, 30, and 3 voxels wide for test cases 1, 2, and 3, respectively. However, the simulation will never reach the maximum chip thickness since the tool is only 24% radially immersed. Using the sinusoidal approximation of the chip thickness, the maximum chip thickness for all 3 test cases should be ~0.11 mm.

Table 2. Parameters for the milling scenario for the voxel size test cases.

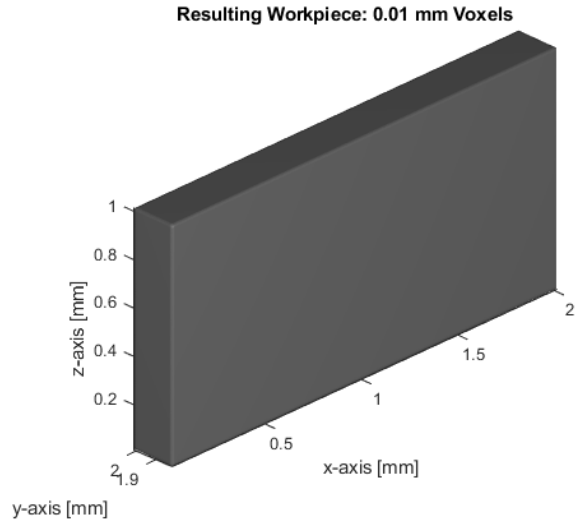
D [mm]	ψ [º]	α_{rake} [º]	SS [rpm]	f_t [mm/tooth]	$r_{imm.}$ [mm]	$a_{imm.}$ [mm]
15	30	0	1500	0.3	1.8	1

The results of the voxel variation test are shown below in Table 3. The model is designed to over remove by a voxel rather than under remove so that if there is a bias in the data it is such that the forces are predicted to be higher. This engineering factor of safety is evident from the results that show for each voxel size the prediction of the critical dimension is offset to be lower than the nominal value of 0.2 mm for a perfect material removal by one voxel width. This shows that while accuracy does decrease with voxel size, the factor of safety and simulation continue to work as designed. Figure 14 shows the remaining solid material after the virtual machining process. The curvature of the workpiece edges is only the result of the plotting functionality in applying the isosurface.

a)



b)



c)

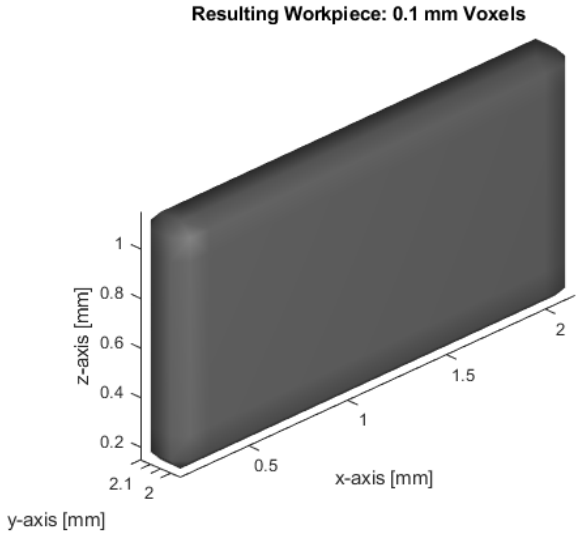


Figure 14. Resulting Workpieces for the of the voxel variation test scenarios

Table 3. Results of the simulation for each voxel size test case. Part width remaining represents a critical part dimension that may need to be measured and checked for tolerance.

Test Case	1	2	3
Mean Part Width Remaining [mm]	0.1975	0.1900	0.1000
Percent Error with Nominal Dimension	1.25%	5.00%	50%

In a parallel to real-world machining the appropriate use of tolerancing in the specification of the simulation will allow for the selection of a voxel size that results in a virtually-machined workpiece that is within the tolerance. However, the appropriate selection of voxel size is further complicated by the use of the simulation to predict milling forces.

In the proposed model the accuracy of the prediction of milling forces is largely dependent on the accuracy of predicting the instantaneous chip thickness shown in

Figure 15.

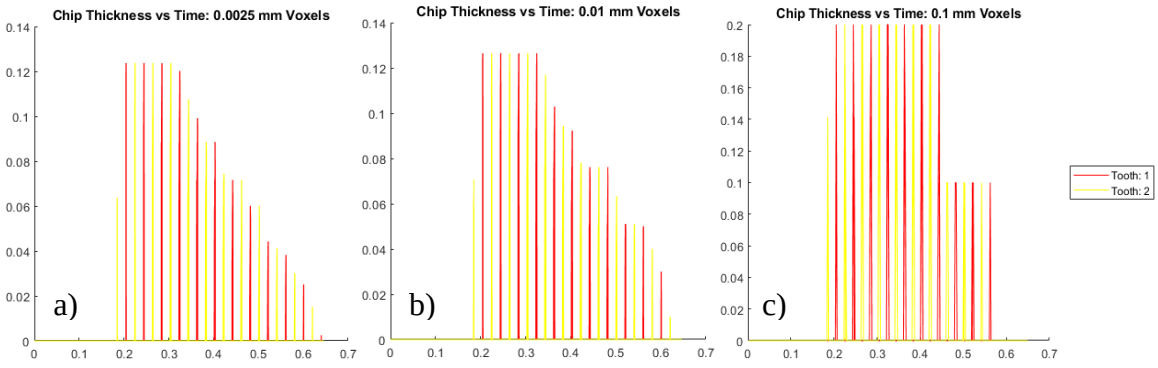


Figure 15. Undeformed chip thicknesses (mm) vs time (s): a) the 2.5 μm voxel size, b) the 10 μm voxel size, c) the 100 μm voxel size.

Therefore, voxel size must be sufficiently small to allow be able to differentiate between the thickest and thinnest portions of the uncut chip thickness as show in Figure 16.

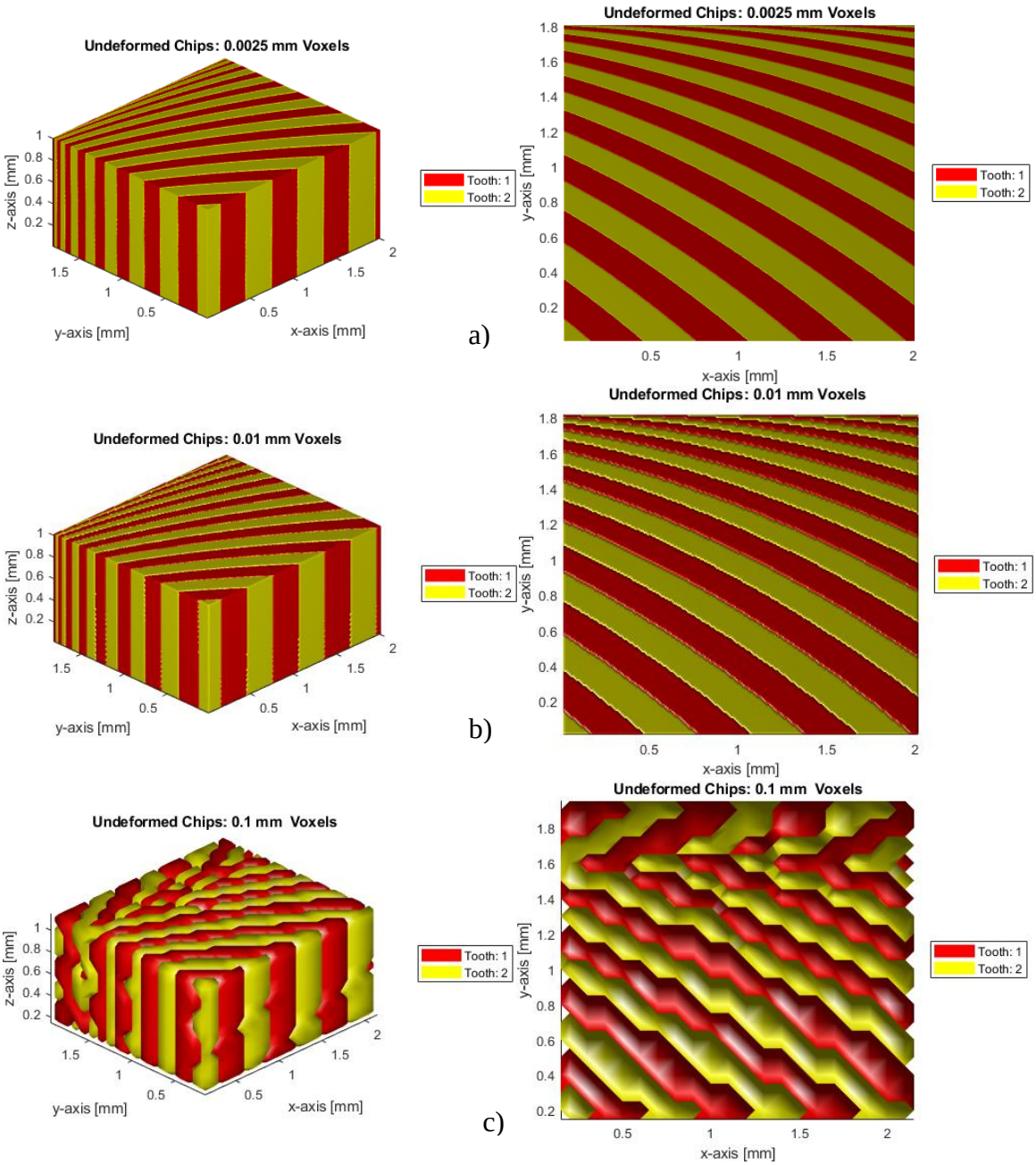


Figure 16. The undeformed chips swept out by the rake lines of the tool for the voxel variation test cases. a) Chips estimated for the 2.5 μm micron voxel size. b) Chips estimated for the 10 μm voxel size. c) Chips estimated for the 100 μm voxel size.

The deterioration of the smooth chip thickness approximations found in Figure 15 a) and Figure 16 a) into the highly aliased data found in Figure 15 c) and Figure 16 c), shows that

the simulation is relatively robust to the variation in voxel size for visualization of the chip formation. As the maximum chip thicknesses vary by only ~0.2 mm or ~ 14.5%, this does not lead to concerns with the visualization of the machining process. However, the results of the force data shows that the increasing voxel size not only causes over prediction of the cutting forces, but also the deterioration of the force signature unique to the period and immersions of the machining operation.

Fortunately, for most milling operations only the prediction of the maximum forces is critical. The accurate representation of the chip thickness near the thinnest portion of the chip is not as critical as the representation of the maximum uncut chip thicknesses. As such, representing changes in chip thickness around the thickest portion of an uncut chip is generally a stronger constraint on voxel size. On the other hand, using voxels that are too large to allow for a clear difference between the thinnest and thickest regions of the uncut chips will result in force predictions that are incapable of accurately predicting the forces as they approach zero. This is due to the clipped range of chip thickness data. Figure 17 and Figure 18 show this loss of accuracy in the lower end of the force data.

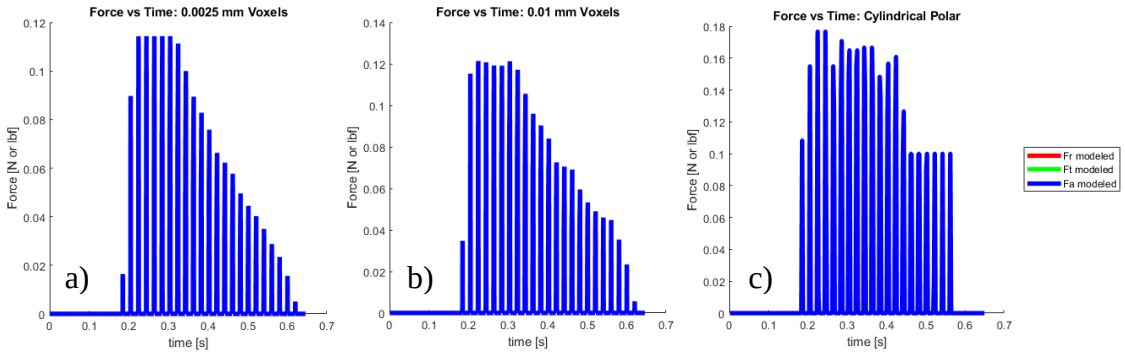


Figure 17. Forces of the voxel variation test cases in the tool coordinate system: a) the 2.5 μm voxel size, b) the 10 μm voxel size, c) the 100 μm voxel size.

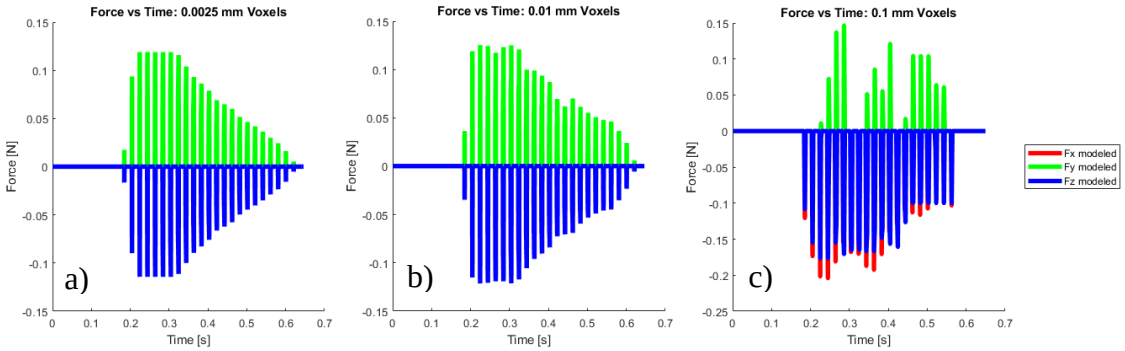


Figure 18. Forces of the voxel variation test cases in the global coordinate system: a) the 2.5 μm voxel size, b) the 10 μm voxel size, c) the 100 μm voxel size.

Inaccurate prediction of the overall fluctuation in milling forces can be detrimental to the milling process. Without a clear picture of how the forces develop between their individual peaks, an assessment of self-excited tool chatter cannot be made. Thus, depending on the milling operation, and the voxel size must be carefully chosen to allow for both simulation efficiency (larger voxels) and data resolution (smaller voxels).

3.1.2 Variation of Rake Line Spacing

The spacing between consecutive rake lines along the flutes of the milling cutter affects the model accuracy and efficiency. Therefore, it was necessary to define an ideal spacing for the rake lines to ensure that the simulation runs accurately and as efficiently as possible. To ensure that the tool is able to correctly remove the workpiece material, it is required that there be at least as many rake lines as there are voxels in the direction of the tool's axis and for the length of the tools axis. Since the tool is designed to be robust to an arbitrary tool specification, the number of rake lines must be able to adapt to the specified tool parameters. Since the sum of the lengths of each leg of a triangle will always be greater than the length of its hypotenuse, the parameters of the tool geometry that require the largest number of rake lines is the case of the simple cylindrical end mill. Therefore,

calculating the number of rake lines required for each flute is as straightforward as dividing the height of the tools flank by the width of the specified voxel size and then adding the value of the tool diameter divided by twice the voxel width specified. The final result is not guaranteed to be an integer so the resulting value is rounded up to the nearest integer amount so that there will always be enough rake lines without generating a far more lines than will ever be needed in the simulation.

3.1.3 Variation of Temporal Step Size

In order to approximate the fluid motion of the cutting tool through the workpiece material, the temporal step size of the cutting tool's motion must be appropriately defined. Since the temporal step size dictates how far the tool moves in the given step of time, it must be driven by a function of the voxel size and machining parameters to remain invariant of the unique machining scenario. Consequently, accurate portrayal of the real-world machining process requires that the tool be capable of removing all of the voxels along the volume swept out by the tools path. Consequently, the constraint on the tool's motion in any given step in time must be that no part of the tool can move more than the width of a voxel in a given time step. Moreover, for 5-axis tool motion even at a constant orientation lead and lean, the motion of the tool cannot be so great as to cause any point of the tool geometry to miss a reasonably sized corner of a voxel. Thus, unlike in a Nyquist-Shannons Theorem [45], the data must be fully sampled and the field exhausted for the simulation to have accurately removed the workpiece volume.

Thus, a point of reference for the tool's motion must be defined as the point of the tool's geometry that moves the most in a given temporal step. For the proposed model of

the tool geometry, this will always be defined as the outermost point along a flute with the greatest radius from the tools axis. With the point of reference defined, the constraint of the tool's temporal step size is simply calculated such that the maximum linear motion, as a combination of the rotation and translation of the tool in a given temporal step, of the reference point must be equivalent to one eighth of a voxel width to adequately capture all of the discrete voxels the tool passes through.

3.2 Validation of the Combined Cutting Model

3.2.1 Validation of Volume Removal

The model was first validated against a simple peripheral milling test where a half inch diameter cylindrical end mill with four cutting flutes at a 30° helix angle was used to remove the workpiece material for the full length of the workpiece geometry. The workpiece was defined as a $^1/_2\ in.\ \times {}^1/_4\ in.\ \times {}^1/_4\ in.$ volume with a voxel size of a hundredth of an inch. The number of the voxels whose value was changed from 'true' to 'false', was multiplied by the voxel-width cubed to obtain the volume of the material removed. From a simple calculation of the prismatic region of the workpiece that was swept out by the tool, it was found that the volume removal of the simulation was exactly that of the back of the envelope calculation for the simple tool pass.

3.2.2 Validation Against Cylindrical End Milling Data

To ensure that the model could correctly predict milling forces, the model was first validated against experimentally measured peripheral milling data for an up-milling operation in Ti_6Al_4V with a cylindrical end mill. This data was collected in Ref. [46] for a

19.05 mm cylindrical end mill of a 30º helix angle, with 4 flutes and a 12º rake angle. The tool was run at 30 m/min in feed with 0.05 mm/tooth of chip load. The tool was set up to be radially immersed by half its diameter with an axial immersion of 5.08 mm for peripheral milling. Table 4 summarizes the milling scenario for this validation test.

Table 4: Summary of the parameters for the first validation against experimental data.

V [m/min]	f_t [mm/tooth]	Radial Immersion	a [mm]
30	0.05	50%	5.08

A rectangular workpiece blank $10.34\ mm\ \times 9.54\ mm\ \times 5.08\ mm$ was generated with a 0.005 mm voxel size. For each flute, the cutter was segmented along the length of the helix into 1,954 steps of equal arc length. As an extra precaution, the simulation was run for multiple additional revolutions after moving into its full engagement with the workpiece. A half second of this fully engaged tool motion was simulated and decomposed into 32,504 discrete time steps. The cutting force coefficients used in this simulation were the same as reported in Ref. [46] and are shown in Table 5 below.

Table 5. Cutting force coefficients used in the validation of the simulation against experimentally measured data for peripheral milling with a cylindrical end mill reported in [46].

K_{rc} [N/mm^2]	317	K_{re} [N/mm]	44.5
K_{tc} [N/mm^2]	1731	K_{te} [N/mm]	22.7
K_{ac} [N/mm^2]	623	K_{ae} [N/mm]	2.4

As seen in Figure 19, the agreement between the predicted milling forces and the forces measured in the experiment is good. It is clear that the modeled machining forces compared well to the measured values both in terms of the magnitude and periodicity of each of the force components.

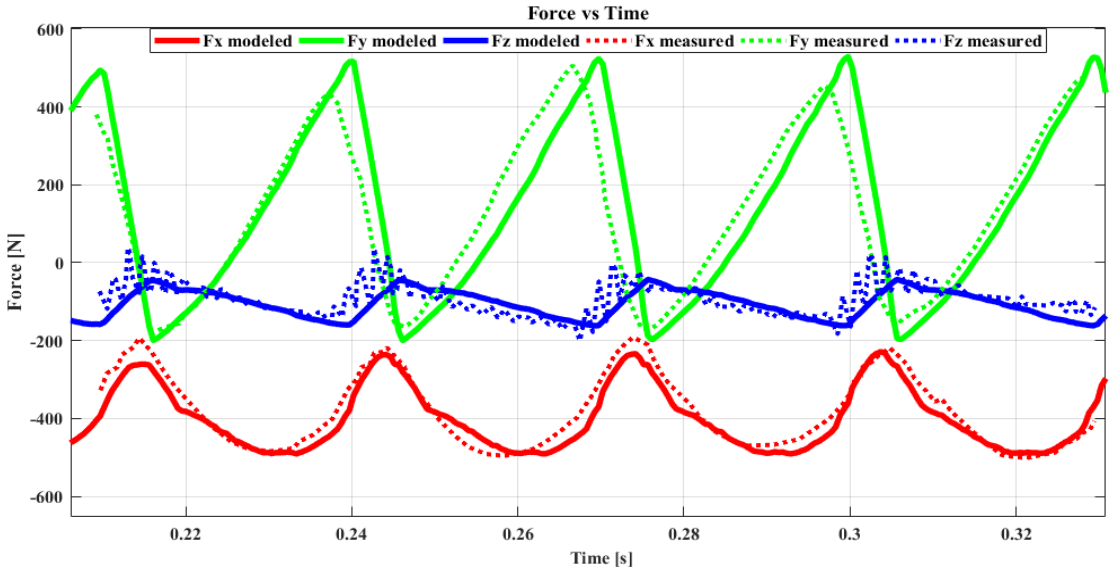


Figure 19. Comparison of milling forces calculated with the voxelized tool-workpiece interaction model and forces measured in [46]. Up-milling with a 19.05 mm cylindrical end mill with 5.08 mm axial engagement of the periphery of the mill and 50% radial immersion.

There are small misalignments in the phase of the signal near the middle peak of force trace for the y-direction; however, this may be due to slight deflections of the cutter during the test. Given that similar misalignments of the force trace for the y-direction are found with the analytic simulation in the literature [14], this seems like a reasonable assumption. Additional minor variations are seen in the force along the feed-direction and the z-direction. These variations in the z-direction occur as the cutting tooth exits the workpiece material and are small in magnitude.

3.2.3 Validation Against Ball End Milling Data

To show that the proposed model also worked for curved workpiece geometry, the model was also validated against experimental down milling data with a ball mill for $AlMgSi_{0.5}$ workpiece material. The data validated against was presented in Ref. [20] for an 8 mm diameter ball mill, with a 30º helix angle and a 12° rake angle. The 2 fluted ball mill run at 4000 rpm was radially and axially immersed by half of its diameter. The parameters of this validation are summarized in Table 6 below.

Table 6. Summary of the parameters for the second validation against experimental data.

V [m/min]	f_t [mm/tooth]	Radial Immersion	a [mm]
0.64	0.08	50%	4

The workpiece was generated to be a $4.65\ mm\ \times 4.01\ mm\ \times 4.01\ mm$ rectangular blank with a 3.75 μm voxel width. The tool motion was subdivided into 1620 rake lines for each flute with 177496 discrete time steps. This allowed for the simulation to run for one full rotation of the cutter after it exited the workpiece to ensure that there was no longer any contact between the cutter and the workpiece at the end of the simulation. The values of the milling force coefficients used are taken directly from figures in Ref. [20] and recorded in Table 7.

Table 7. Cutting force coefficients used in the validation of the simulation against experimentally measured data for down-milling with a 50% radial and axial immersed ball mill reported in [20].

K_{rc} [N/mm^2]	125	K_{re} [N/mm]	6.25
K_{tc} [N/mm^2]	575	K_{te} [N/mm]	12.5

K_{ac} [N/mm^2]	150	K_{ae} [N/mm]	1

Figure 20 shows the results of the simulation compared with the results of the cutting experiment. According to the author of Ref. [20], the discrepancies in the magnitude of the force along all three directions is primarily due to cutter runout in the experiment.

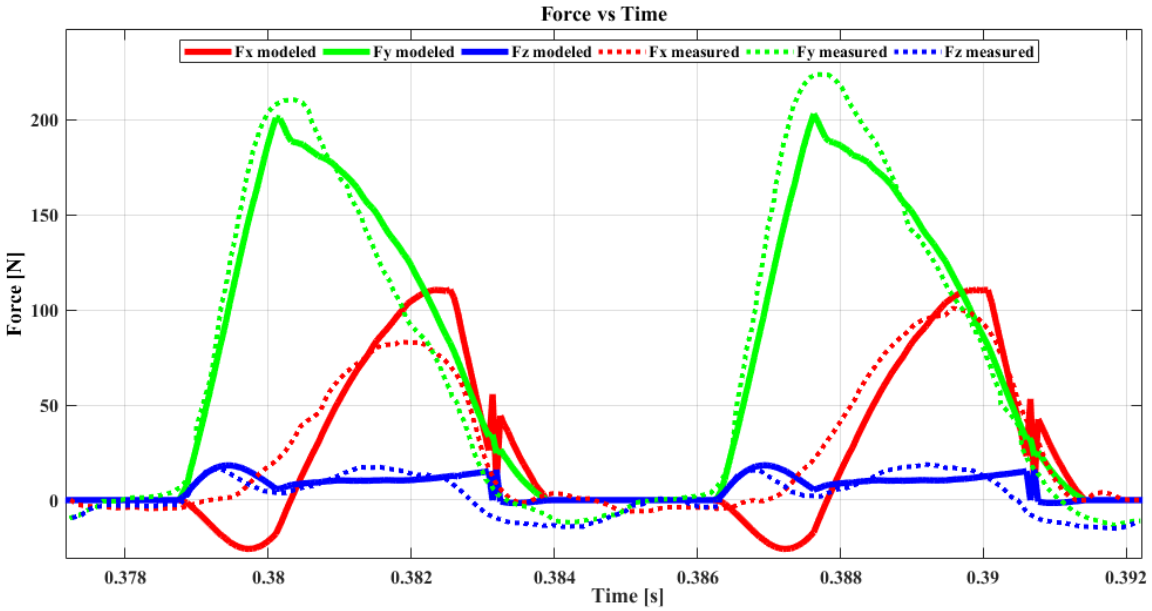


Figure 20. Comparison of the down-milling forces for an 8 mm ball mill in $AlMgSi_{0.5}$ sample.

Despite some lead and trailing edge fluctuations along the x-direction, the results show a good agreement with the measured force trace from the experiment. The initial sink in the forces along the x-direction is characteristic of using constant force coefficients for the length of the ball mill and can be corrected with a distributed cutting force coefficient relationship. The fluctuations in the predicted x force as the flute exists the material are not noticed in simulations with higher numbers of flutes or where the ball end of the mill is not fully immersed it is likely that they may be due to infinitesimal chip thickness at the trailing edge of the chip in down-milling that can be drowned out when more cutter teeth are

making contact with the workpiece material. Moreover, this systematic error would likely decrease with decreasing voxel size.

CHAPTER 4. SIMULATION AND RESULTS

4.1 Challenges in Representing Unique Milling Scenarios

While there are ample models in research today that can accurately predict milling forces for the whole realm of cutting tools being used in industry, most practical models are based on sinusoidal approximations of the cutting flute's contact with the workpiece material without any prior knowledge of the workpiece surface that is being cut. Existing models that rely on cutter-workpiece intersections for surface geometry rely on complex definitions of engagement bounds that require the calculation of intersection points with floating point precision. In the cases of non-cylindrical tools or non-prismatic workpieces, these methods analytical solutions break-down and reliance on spline formats to define the engagement regions [28]. Once the equations cannot be solved, numerical solvers are used before preforming the integration. As such, these methods are unable to nimbly and efficiently represent the interaction of the milling cutter with the workpiece in complex engagement situations. This is due to the general uniqueness of modern machining paths with multi-axis tool paths that go beyond 3-axis, cylindrical tool, and prismatic part machining.

4.1.1 General Uniqueness of Tool Passes

As CNC machine tools become more common in even small machine shops, advanced tool paths are becoming the standard. The specific reasons for these complex paths are numerous, but include improved material removal rates, chip thinning, and insulating against chatter or thermal effects. CAM software has made the application of

complex tool paths extremely easy with adaptive clearing and similar approaches. While these tools are great for improving the productivity of our machining centers, they bring with them tool paths and subsequently tool-workpiece engagements that are impractical to analyse with traditional models. This problem is exacerbated when 5-axis path planning is considered; rendering sinosoidally approximated chip thickness models and models that rely on calculating intersections between surface geometries inadequate. It is precisely due to these drawbacks that there is a push in some literature and companies now for voxel based workpiece representations and path planning CAM software [33, 32, 47].

4.1.2 Complex Interactions in Multiple Pass Machining

Traditional CAM softwares fall short in terms of the computational complexity in computing Boolean operations for constructive solid geometry and boundary representations [48]. In addition, without the complete representation of the tool geometry CAM softwares are unable to predict the interaction between the tool and workpiece beyond the Boolean subtraction operations. With a more complete representation of the tool geometry it should be possible to incorporate the effects of the previous tool path on the workpiece conditions for the following operations. One down side of aggressive roughing is that the resulting workpiece is often left in a state of high surface roughness that can lead to unexpected tool deflection or chatter in following passes. Since they rely on envelope cutter geometry, this surface roughness is not something traditional CAM methods are able to model. As a consequence, the ability to work with aggressive roughing passes is limited. With additional information about workpiece resulting from the previous tool pass available, improvements can be made to the entire machining process to improve both speed and accuracy.

4.2 Parameters of the Two-Pass Simulation

The model presented allows for the collection of this additional workpiece information without any increase in the complexity of the original workpiece model. This advantage makes this model perfectly suited to analyse the effects of previous tool passes on subsequent ones. Outlined in this section are the parameters, results and discussion of a couple of two-pass simulation scenarios. Two scenarios are developed to examine the effects of previous roughing tool passes. Both scenarios use a 30 mm diameter 2 fluted roughing tool followed by a 5 mm 4 fluted finishing tool. In addition, the conditions for both finishing passes are kept consistent to avoid misappropriation of the differences in the results. The finishing cutter flutes are inclined at a helix angle of 60° and 15° rake angle. The finishing passes are run at 6000 rpm with 0.1 mm/tooth of chip load. These parameters are summarized in the lower half of Table 8 and Table 9 below. The roughing cutter for the first two-pass simulation scenario is run at 1000 rpm for a chip load of 1.5 mm/tooth. In addition this cutter's helix angle and rake angle are also 60° and 15° respectively. The parameters for the first roughing cutter pass (scenario 1) are summarized in the top portion of Table 8.

Table 8. Simulation parameters for scenario 1.

Diameter [mm]	V [m/min]	f_t [mm/tooth]	Radial Immersion	a [mm]
30	3	1.5	1/5 (3 mm)	5

5	2.4	0.1	1/5 (0.5 mm) nominal	5

For the second scenario the roughing cutter shares the same dimensions as the rougher from scenario 1, but the chip load is more than halved at 0.6 mm/tooth to show the effects of feedrate on the protrusion of the helical flute scallops on the surface of the workpiece. The parameters for the roughing cutter in the second scenario of the two-pass simulations are shown in the top half of Table 9.

Table 9. Simulation parameters for scenario 2.

Diameter [mm]	V [m/min]	f_t [mm/tooth]	Radial Immersion	a [mm]
30	1.2	0.6	1/5 (3 mm)	5
5	2.4	0.1	1/5 (0.5 mm) nominal	5

To avoid any biasing of the force results and to better demonstrate the direct effect of the rough surface on the following tool path's chip thickness, the cutting force coefficients are set to 1 for all the shear elements and zero for all the ploughing elements. The ploughing elements are set to zero since their application is not dependent on the calculated chip thickness. The summary of the cutting force coefficients for both two-pass scenarios is found in Table 10 below. By limiting the shear cutting force coefficients to unity, the force prediction of the simulation becomes a direct representation of the effects of the calculated chip thickness in each direction of the workpiece coordinate system.

Table 10. Summary of the cutting force coefficients used in both scenarios of the two-pass simulations.

K_{rc} [N/mm^2]	1	K_{re} [N/mm]	0
K_{tc} [N/mm^2]	1	K_{te} [N/mm]	0
K_{ac} [N/mm^2]	1	K_{ae} [N/mm]	0

4.3 Results of the Two-Pass Simulation

The workpiece resulting from the tool pass of the 30 mm cutter at 1.5 mm/tooth are shown below in Figure 21.

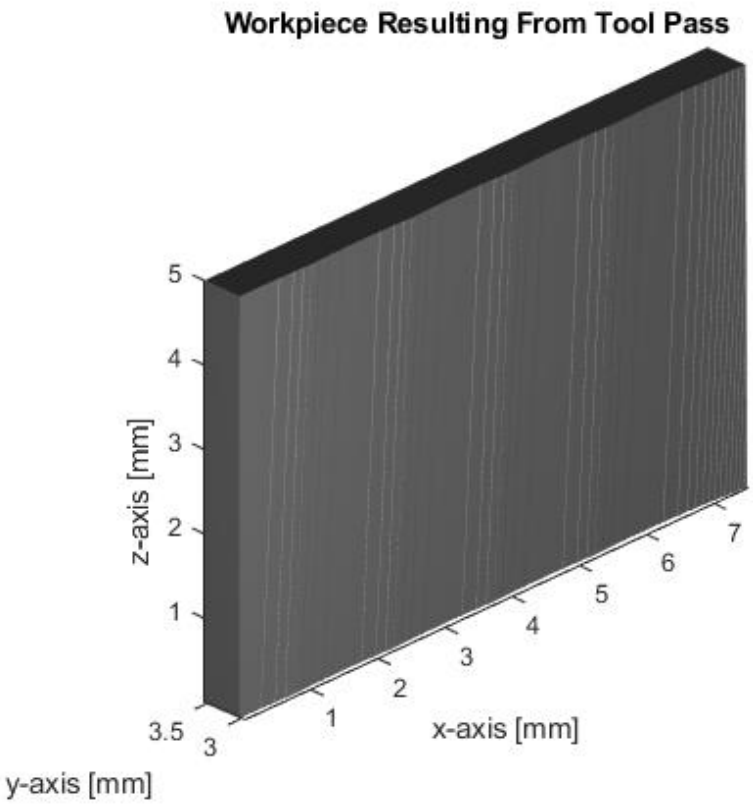


Figure 21. Workpiece resulting from the motion of the 30 mm tool run at 1.5 mm/tooth. The wavy surface of the workpiece is a direct result of the high feedrate relative to the fixed spindle speed for this tool.

The high feedrate of this tool pass has left the surface of the workpiece rough with the periodic scallops cresting above the planned workpiece surface. These scallops exist in all machining passes, but are often insignificant to the evaluation of the dimensions in more conservative passes. This is due to the motion of each cutting flute moving through the workpiece in a trochoidal path that will always result in a non-infinitesimal scallop height [5]. Figure 22 gives a visualization of the uncut chips that are swept out by the cutting flute of the roughing tool in this scenario. The high feed per tooth is evident from the thick bands of each uncut chip shown. Near the back left edge of the part, the periodicity of the trochoidal tooth paths can just be made out from the curvature of the chips alternating from flute 1 and 2.

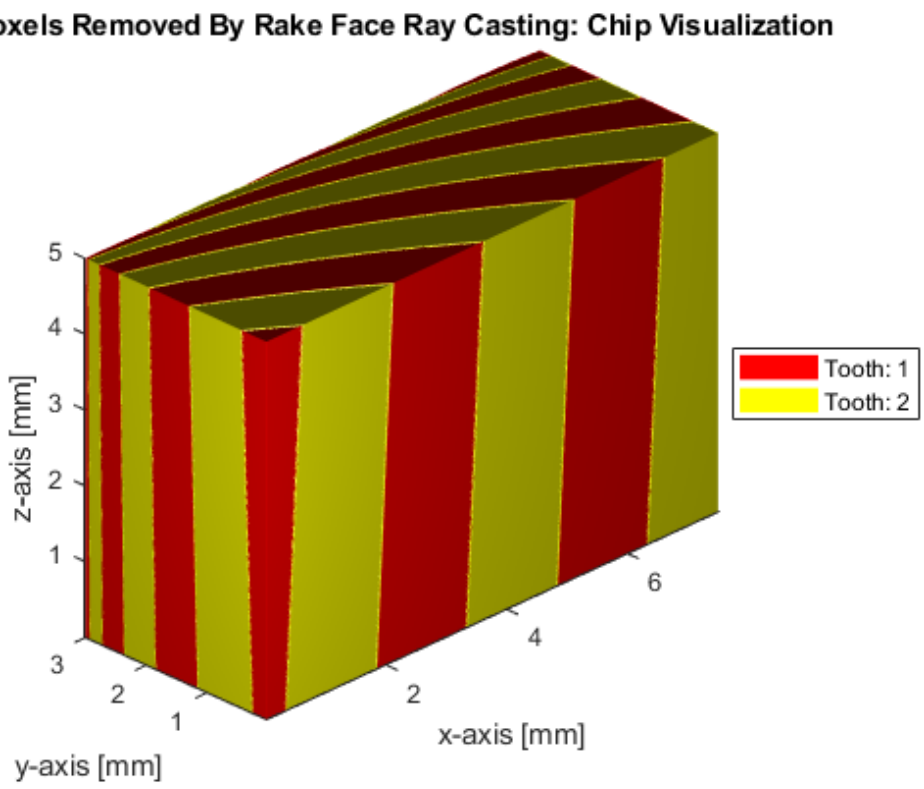


Figure 22. Visualization of the uncut chips removed from the workpiece by the 30 mm tool run at 1.5 mm/tooth.

As the roughing cutter moves through the workpiece along the x-direction, the different portions of the uncut chip are visible. From 0 to approximately 3 mm, the region of the tooth's path through the workpiece generates the thicker region of the chip, whereas, the chips generated from approximately 4 mm to the end of the workpiece are the thin beginnings of the uncut chips for up-milling. This means that the type of chip load the cutter sees during its engagement with this short workpiece length changes throughout the process. This change is clearly visible in the forces predicted below in Figure 23.

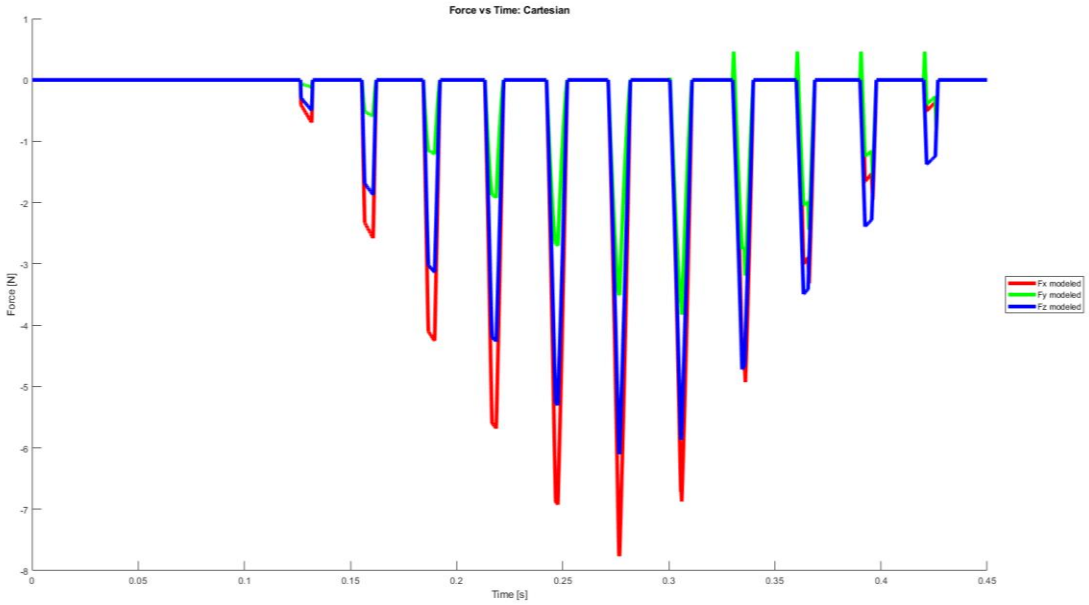


Figure 23. Forces predicted for the 30 mm tool run at 1.5 mm/tooth.

The transformed effects of the calculated chip thicknesses on all three directions of the workpiece coordinate system show that as the tool is in the middle of the tool pass, the force peaks as expected since it is at its longest chip engagement with the workpiece material. As the tool moves into the workpiece at around 0.125 seconds, the trend of the peak forces grows almost linearly with the tool traveling deeper into the workpiece. As the tool nears the latter half of the workpiece, the forces decline asymptotically towards zero with the exception of the force in the y-direction. The force in the y-directs begins to exhibit small peaks of the force in the opposite direction due to the lagging entry of the cutting flute to the position of the center of the cutting tool. This can be thought of as a negative angle for the flute to enter the material relative to the local y-axis of the tool center point. This effect only becomes apparent as the tool exits the part, as the lower region of the tool's helical flute is no longer in contact with the workpiece material at what could be considered a positive angle to the local y-axis of the tool center point. This results in a lack of sufficient force in the original direction of the y-forces to suppress the positive forces from becoming

apparent. This switch of force direction is a real factor in everyday machining as the cutter approaches the exit of the machining path, but is rarely noticed due to lower chip loads and the scaling of cutting force coefficients.

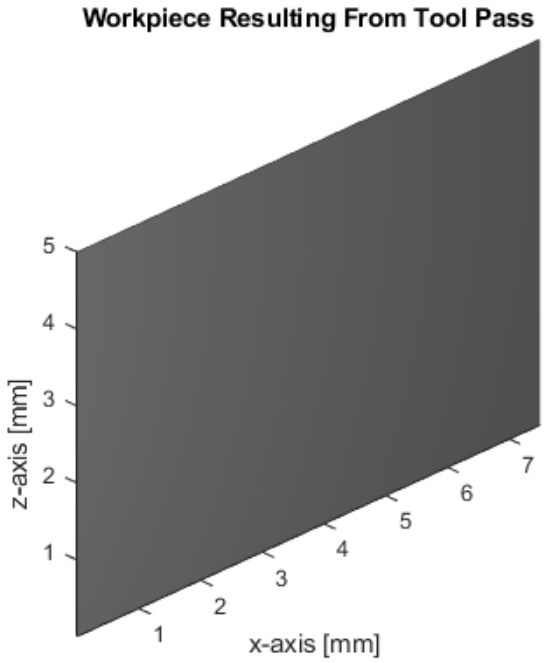


Figure 24. Workpiece resulting from the motion of the 5 mm tool run at 0.1 mm/tooth.

Figure 24 above shows the workpiece after the finishing tool pass for the first scenario. The noticeably smoother surface is to be expected with a finishing cutter since it is run at an order of magnitude lower chip load with a higher spindle speed. The overall result is that the part surface is near smooth. The smaller scallops left by this tool pass are considered small enough to be negligible for the sake of this analysis. Below in Figure 25, the visualization of the uncut chips swept out by the finishing cutter are shown. For the sake of consistency, the uncut chips are shown at the same angle throughout these simulations, although this limits the perspective of the image on the periodic surface scallops of the scenario one roughing pass.

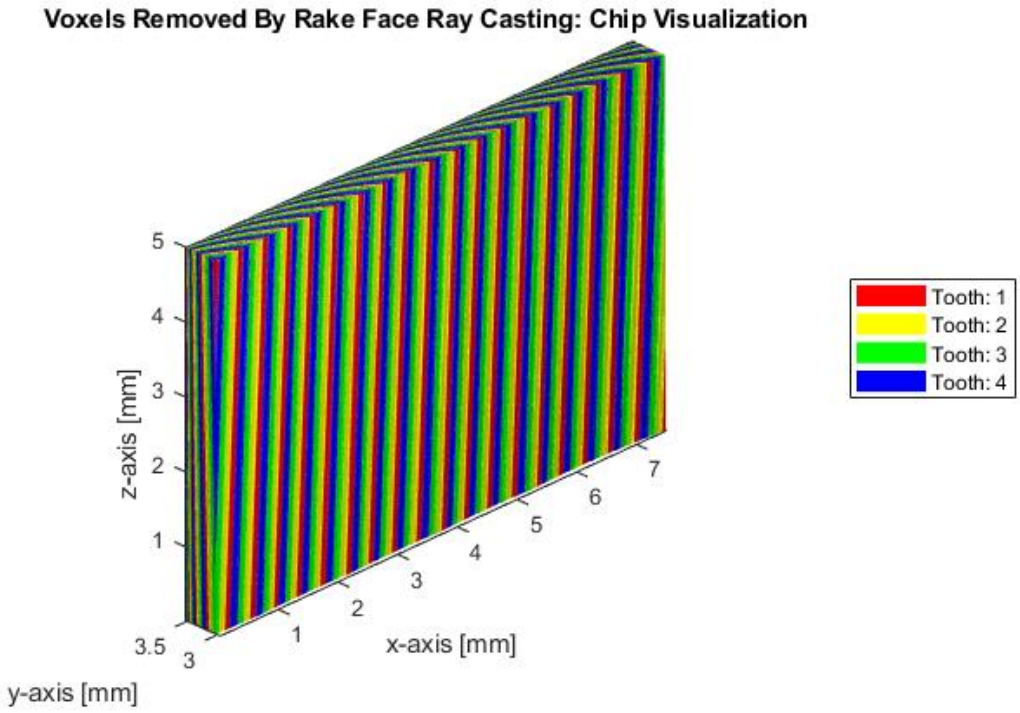


Figure 25. Visualization of the uncut chips removed by the 5 mm tool run at 0.1 mm/tooth.

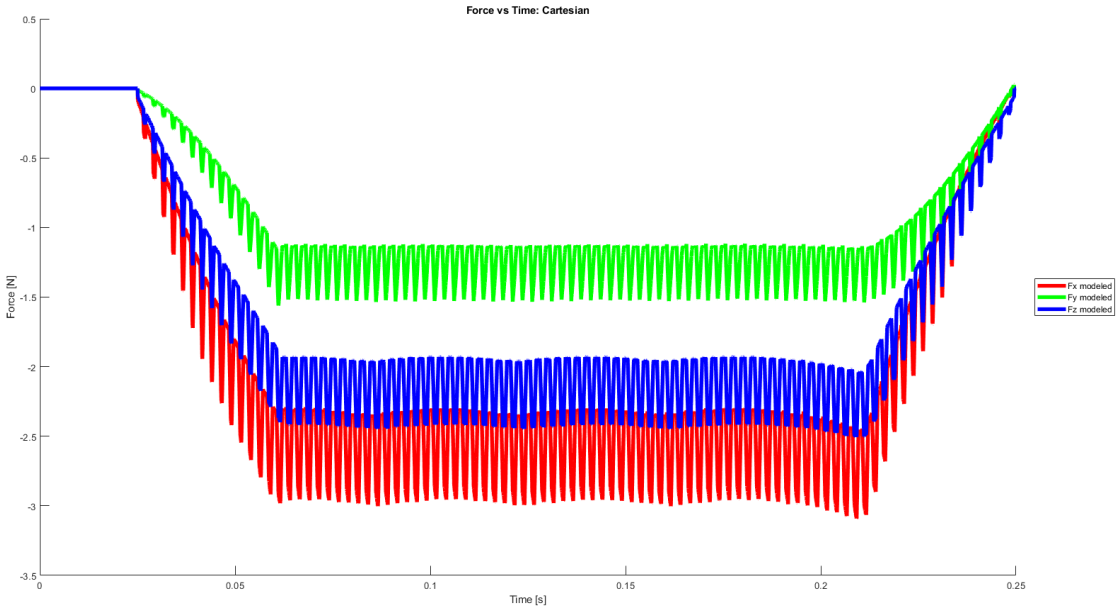


Figure 26. Forces predicted for the interaction between the 5 mm tool run at 0.1 mm/tooth with the workpiece resulting from the 30 mm tool run at 1.5 mm/tooth.

Under close inspection, it can be seen that these surface scallops were picked up by the finishing tool and result in the increase of its radial immersion of this finishing tool up to the crest of the scallop. This effect is most clearly seen in the undulation of the predicted

forces shown below in Figure 26. The periodicity of the scallops left by the roughing tool pass for this scenario are clear from the fluctuations of the forces shown above. The lack of large fluctuations in the forces along the z-direction is attributable to the independence that the z-forces have from the angle of the about the tool axis and, as such, is only dependent of the angle κ from the tool axis as described in equation (59).

The higher frequency of the periodic surface scallops in Figure 27 is the result of running the roughing tool for the second scenario at the same spindle speed as the first, but with a lower feedrate. This reduces the spacing between the scallops left by the cutting flutes on the surface of the workpiece, while also reducing the scallop height from the nominal workpiece surface. As can be expected, this results in a more precise workpiece surface at the cost of a lower material removal rate. Figure 28 shows how this lower feedrate relative to scenario 1 allows for the cutter to experience smoother chip development through the milling process with the region of the uncut chip transitioning more smoothly from the front of the workpiece (thick chip region) towards the back of the workpiece (thin chip region).

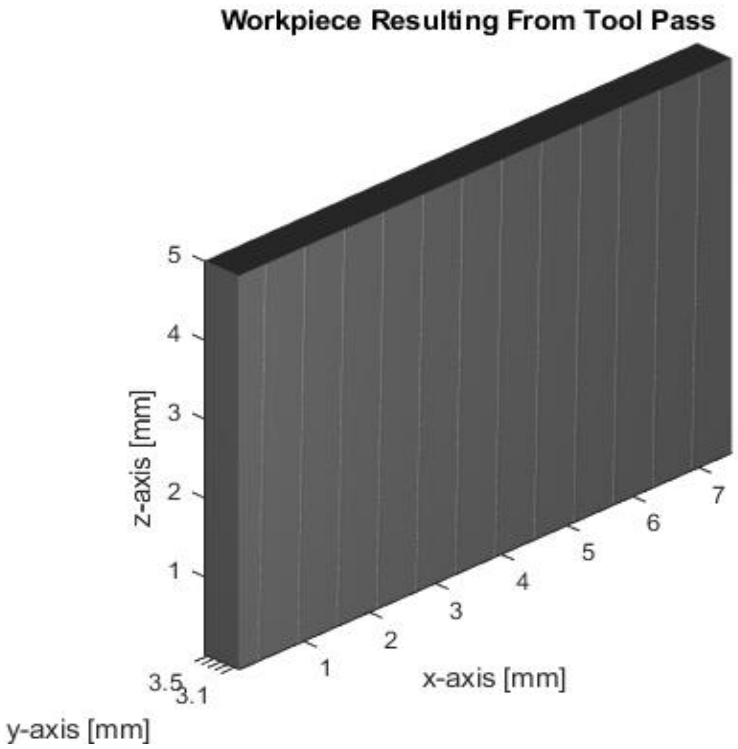


Figure 27.Workpiece resulting from the motion of the 30 mm tool run at 0.6 mm/tooth. The higher frequency of the wavy surface is due to the slower feedrate of this scenario.

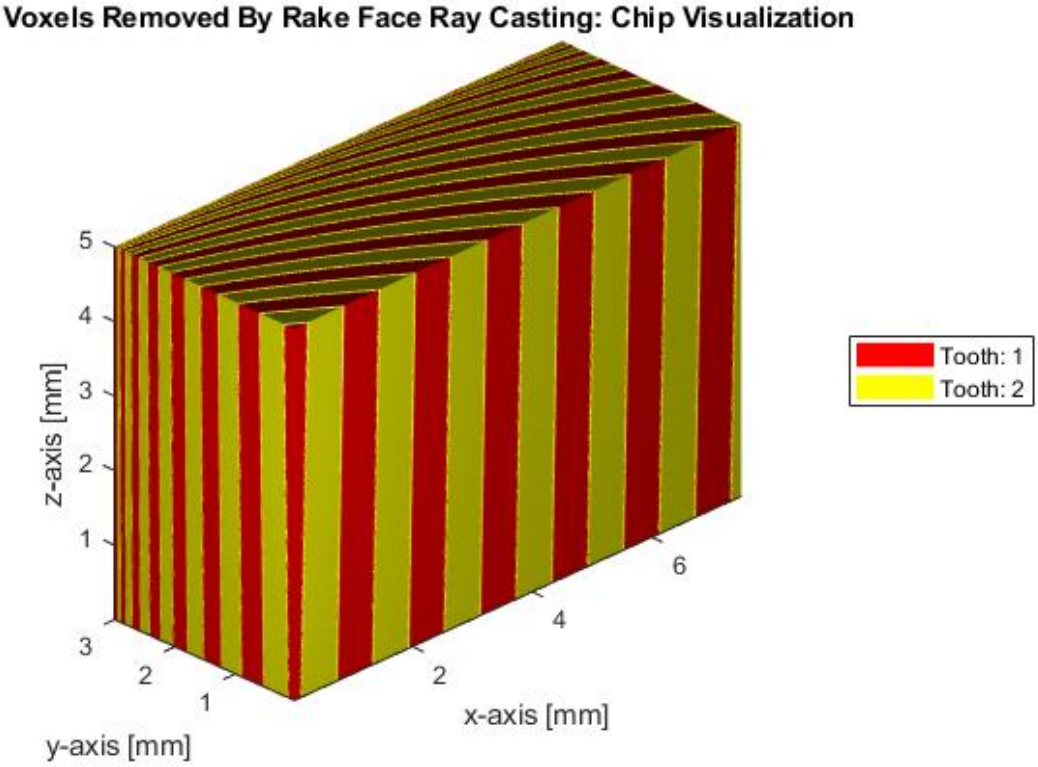


Figure 28. Visualization of the uncut chips removed from the workpiece by the 30 mm tool run at 0.6 mm/tooth.

This improved transition is seen in the predicted forces with a more normal distribution than the previous scenario. Otherwise these forces, shown in Figure 29, display similar behavior to those for scenario 1 in Figure 23.

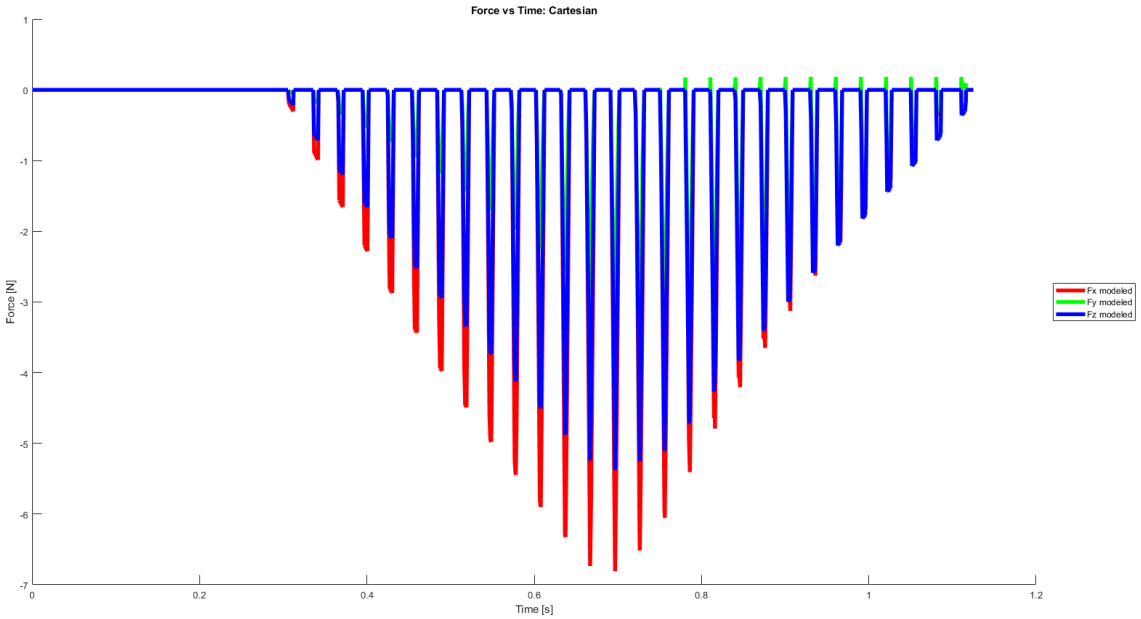


Figure 29. Forces predicted for the interaction of the 30 mm tool run at 0.6 mm/tooth with the workpiece.

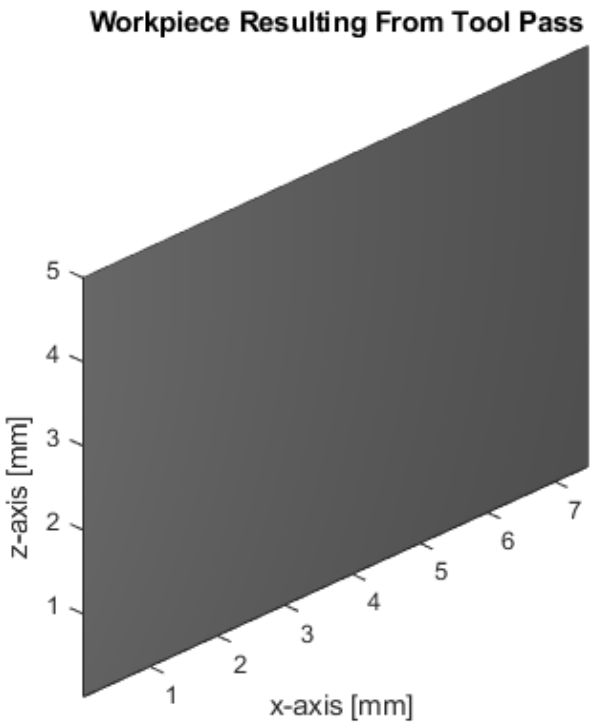


Figure 30. Workpiece resulting from the 5 mm tool run at 0.1 mm/tooth.

Once again the workpiece resulting from the finishing tool pass is significantly smoother than the workpiece resulting from the roughing tool pass, and has negligible surface roughness for the sake of this analysis. While the frequency of the finishing cutter, easily seen in Figure 31, is held constant between the two scenarios the frequency of the roughing cutter causes this scenario to collide with scallops more frequently, but with less magnitude. The results on the force prediction are shown in Figure 32.

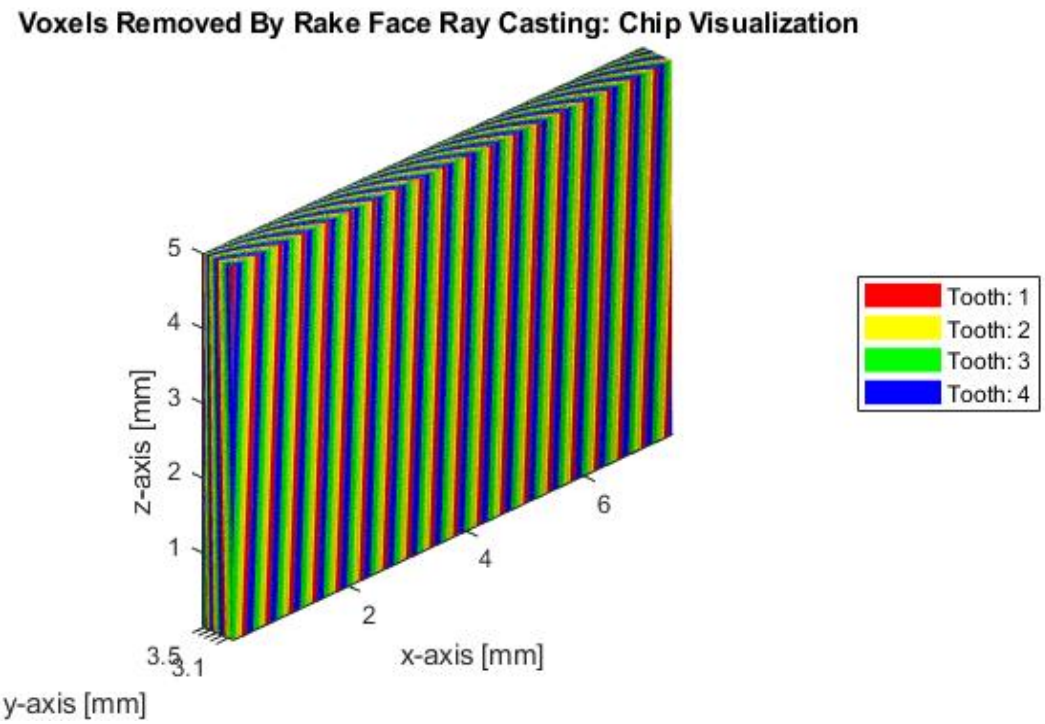


Figure 31. Visualization of the uncut chips removed from the workpiece by the 5 mm tool run at 0.1 mm/tooth.

Due to the decreased surface scallop height and the period of the finishing cutter, the effects of the roughing tool pass in scenario 2 are greatly diminished from those in scenario 1. The slight undulation of the forces in the x and y directions peaks occur approximately every 6 flute passes of the finishing cutter. This constructive interference in the finishing cutter force prediction is relatively minor compared to the first scenario of the two pass simulation. This lessened effect is attributable to the decreased chip load of the roughing cutter in scenario 2 as well as the interference from the roughing tool pass which falls on the sweep of each cutting tooth of the finishing cutter.

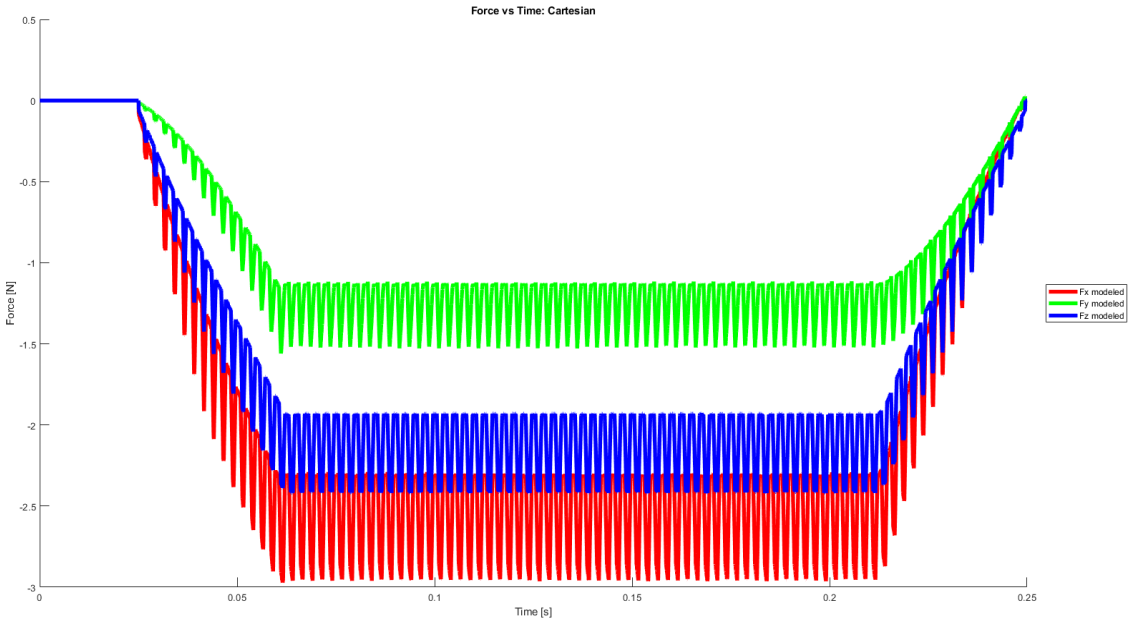


Figure 32. Forces predicted for the interaction between the 5 mm tool run at 0.1 mm/tooth and the workpiece resulting from the 30 mm tool run at 0.6 mm/tooth.

4.4 Discussion of the Simulation Results

It is clear from the simulations on the effects of the variation of voxel size that there are limits to the model's accuracy, as the voxel size approaches the order of magnitude of the maximum chip thickness due to the feedrate of the cutting tool. As these limits are approached, accurate prediction of the cutting force falters at an earlier stage compared to accuracy of critical part dimensions of the virtually machined workpiece. This is primarily due to the breakdown of the calculation of the instantaneous chip thickness across the length of each chip. As stated above, voxel size must be sufficiently small for the accurate estimation of chip thickness at several locations distributed across the length of the chip. This relationship is clear in Figure 33, as the cutting flute moves through the material in a trochoidal tooth path.

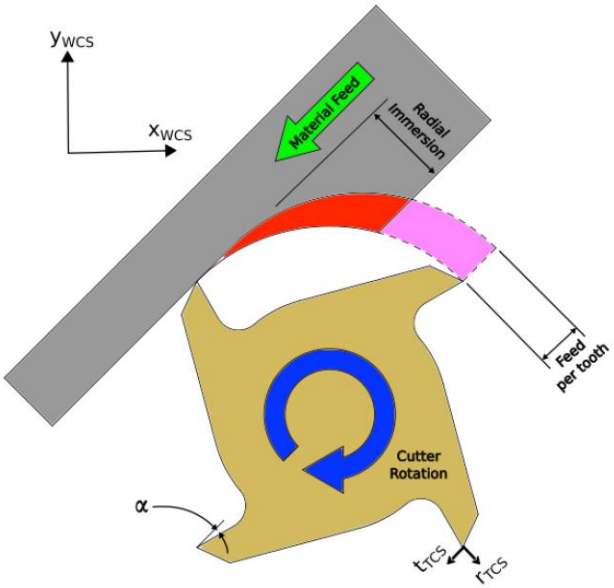


Figure 33. Illustration of how chip thickness evolves along the length of the chip for an up-milling operation.

The volume removal tests showed that for medium-to-large voxel sizes and low chip loads, the model can be validated against simple prismatic calculations of the removed material volume. For smaller voxel sizes and high chip loads, the model is able to predict the volume removal more accurately than standard CAM solutions and the prismatic calculations.

The validation test against the experimental data on peripheral milling with a cylindrical end mill collected by Budak in Ref. [46] shows that it is appropriate to apply a mechanistic model designed for simplistic sinusoidal chip thickness estimations to the discrete data produced by this model. Furthermore, by accurately predicting the force data produced by a specific cutter in the experiment, it is shown that the model is able to sufficiently simulate the interaction of the cutting tool with the workpiece volume through considerations of only the rake face of the cutting tool. This is important as it validates the

assumption that the tool geometry can be reduced to rake face geometry for the sake of the efficiency of the simulation.

The data sourced from Budak et. al showed that the model was capable of predicting forces, but for the sake of completeness, validation against down-milling operations was necessary. The additional test against the down-milling of $AlMgSi_{0.5}$ showed that the model was further capable of simulating unusual milling operations such as the full diametrical immersion of the ball portion of a ball-mill. While the data collected by Gradiŝek et. al was not as free of cutter run-out as the Budak experimental data, it was considered important that proposed model still was able to show agreement. This is due to the fact that small discrepancies like cutter run-out are more common than not in most milling tests and real world milling operations. This is due to the general robustness of the milling operation due to the effective radial symmetry of the instantaneous tool motion about the spindle axis, assuming spindle speed is much greater than the feedrate. The irregularities of the predicted force and the force measured along the x-direction are interesting as lesser axial immersions of the ball mill with the same radial immersion, feed, and speed showed the same behaviour in the measured force in the x-direction as does the analytical model based on constant cutting force coefficients in Ref. [20]. However, the analytical model depth dependent cutting force coefficients were able to more accurately predict this measured force. This implies that the proposed model can be further improved by applying a similar regime of immersion-dependent force coefficients. Furthermore, the model already contains the requisite data for such applications since all of the points in space are already known. This means that for non-trivial immersions, such as depth of cut in prismatic workpieces, more akin to those found in 5-axis freeform surface machining, the model

would still be able to apply the correct cutting force coefficients. This is beyond what has been done with analytical models to this point.

The fluctuation in the predicted results, as shown in Figure 20, shows the drawbacks of the discrete nature of the workpiece data. While proper sampling of the chip thickness data along the temporal steps of the simulation does help to smooth the data to help estimate the continuum behaviour of the material, it is not a perfect solution. This is due to the fact that the data is also discretized along the axis of the tool. Additional sampling along the each flute of the tool could help to improve this behaviour, but doing so would limit the resolution of the results from the unique sections of more complex tool's geometry. This limitation is especially important when questions of immersion arise with tapered bull-nose cutters or ball vs. flank milling in ball-mill cutters.

Complex tool paths are an important and growing portion of the modern machining industry as these paths grow in complexity traditional machining knowledge and analytical machining models are no longer capable of providing the kind of analysis and insight into the machining process as is needed. This means that a lengthy trial and error optimization phase is now default for many complex part industries. This is of course costly and inefficient in a market that is only growing more competitive as a large portion of the world develops their own manufacturing industries. The need for models capable of handling complex tool and workpiece interaction while also being robust enough to handle any number of machining scenarios is substantial. The full range of 5-axis motion and the ability to simulate arbitrary tool geometry pose the developed model to be able to address these growing needs of industry.

The ability to simulate consecutive tool passes sets the proposed model apart in that it is an inherent advantage of the chosen data structure over both analytical and surface based models. Both of which become more complex the more complex the part surface becomes. With the predefined resource demands of the workpiece data and the efficiency of the integer based voxel traversal scheme, there is little to no increase in computational demand for the proposed model. Moreover, since the surface based techniques require numerical solvers for the equations of the engagement for most non-prismatic situations, this model's computational structure is better suited for complex machining scenarios.

The data in Figure 21 and Figure 27 show that this model is capable of detecting the effects of previous tool passes and predict how they will affect the current pass. There is however a limitation on this ability to detect the results of a previous tool pass. This limit is directly related to the relationship between voxel size, tool diameter, and chip load. For linear tool paths the distance between the lobes of the trochoidal motion of the cutting teeth is the same as the chip load [28]. Since the chip load is always significantly smaller than the diameter of the cutting tool for milling operation, the local path between any two consecutive lobes can be approximated as linear. Therefore, a simple relationship can be defined for the ratio of voxel size to tool diameter and chip load [49].

$$w < \frac{D}{2} - \sqrt{\left(\frac{D}{2}\right)^2 - \left(\frac{f_t}{2}\right)^2} \tag{62}$$

Equation (62) shows that for the simulation to be able to detect the scallops resulting from a tool path the voxel size must be significantly smaller than maximum height of the scallop from the new nominal surface of the workpiece. The particular ratio is still dependent on

the tolerances of the part being made, because given the cubic geometry of the voxel the infinitesimal peak of the scallop cannot be simulated. So a reasonable compromise can be found with a simple examination of the required geometric tolerances of the part being made. It can also be seen in Figure 26 and Figure 32 that detection of the surface roughness is not enough to clearly show how it will affect the force predictions of the successive pass. The relationship defined in equation (62) still holds, but the additional constraint of the voxel size being sufficiently smaller that the following tool passes chip load must now be the dominating factor.

Due to the greater magnitude of other common variations in the forces measured when cutting metals, the variation in the cutting forces from the scallops resulting from the trochoidal path of the previous cutter's teeth are generally not significant enough to concern most machining operations. However, by showing that this simulation is capable of detecting such minute features and how they affect the subsequent pass across the surface, this model is shown to be more than capable of predicting the influence of macro scale scallops due to step over of curved mill geometries such as those that commonly occur in freeform surface machining and many 5-axis tool paths. This is important because these geometric fluctuations in workpiece geometry mean that the tool passes that follow and run through these scallops are generally run at extremely conservative rates. With the insight provided by this proposed model, these clean-up operations for 3- and 5-axis sculpted surface machining can be improved and optimized so that even small scale production runs not warranting a trial and error manual optimization phase can be optimized by these virtual machining simulations. This has the potential to have a large impact on the aerospace, automotive, mold and die, and defense industries, in that the

production of blisks, compressor blades, molds, and impellers, to name a few, will be made significantly easier.

The results of the subsequent passes in both Figure 26 and Figure 32 show that the detected surface scallops have an effect on the predicted cutting forces in both the x- and y-directions. If the scallops were not essentially aligned with the axis of the cutting tool, as they may not be in 5-axis milling passes, their effect would also be noticeable in the forces along the z-direction. It can be seen that the chip load of the previous tool pass has a direct effect on the scallop's effect on the subsequent pass's predicted forces. As the chip load of the previous tool pass increases, the magnitude and period of influence on the forces of the subsequent tool pass increase. In addition, the alignment of the frequencies of the consecutive tool passes can influence how much the scallops affect the subsequent pass's force predictions. If the head of the chip, meaning the thickest portion of the chip, occurs within the region of a surface scallop the forces predicted for that moment in time will be greater than if the region of the scallop falls in the middle of the chip or near its tail. The latter two circumstances will of course still locally increase the cutting force, but not the local peak cutting force.

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The approach of the model presented is to create a representation of the solid nature of the workpiece and simulate the effective material removal process of the milling tool with that workpiece body. To ensure that the model would be capable of simulating a practical representation of this process, the discrete nature of the both the workpiece and the tool were varied and the limits of the models robustness were defined. The validity of the model was tested against data calculated for simple prismatic volume removal operations and found to be accurate within a tolerance that the can be controlled with the specification of the voxel size. As one step to ensure that the model could accurately predict the effects of real machining scenarios, the model was validated against experimental data for peripheral up-milling of Ti_6Al_4V with a simple cylindrical mill. It was found that despite the discrete nature of the model's framework with a simple sampling regime, the continuum of measured results could be recovered sufficiently to have good agreement with the measured force data from the cutting experiment. The validity of the model was further tested by the comparison of the simulated results of a down-milling operation with a ball-mill that was immersed for the full depth of the ball end of the mill. It was found that the model was able to sufficiently predict the measured behaviour and the analytical model result using the same constant cutting force coefficients.

By extension of the model in terms of axial engagement dependent coefficients, the proposed model could lend an even more in-depth analysis to sculpted surface machining where the ball mill depth will vary greatly across the tool path. Information about the

instantaneous depth of any point on the model is easily obtainable from the information already contained in the model such as the height of the last rake lines engagement with workpiece material for an instance in time and that instance's location of each rake line. As such, the model is uniquely able to adapt easily to complex and unique machining scenarios without any increase in its computational complexity, unlike analytical and surface based models. On this point, the model was shown to be able to predict the effects of prior tool passes and their resulting surface roughness on the subsequent tool pass in both workpiece geometry and predicted forces. While the roughness directly resulting from the trochoidal motion of the tool's cutting teeth was discussed in terms of its generally low significance, these results showed that large surface roughness or varying surface geometry can be easily detected and accounted for in the simulation of consecutive tool passes. The nature of the voxel size to tool diameter and chip load was discussed, and quantified in equation (62), in terms of how well the model can detect and predict the results of these unique elements of the workpiece geometry that result from the machining process. The nature of the relationship between the frequencies of the consecutive tool passes was discussed, and the constructive interference of the scalloped region of the force predictions of the subsequent pass at that location were described.

5.2 Recommendations

It was shown that the higher chip load of the roughing tool in the first two-pass simulation scenario caused a longer period of raised forces in the subsequent finishing tool pass than the lower chip load of the rougher in the second scenario. This implies that the larger the ratio between the diameter and chip load of the previous tool and those of the current tool, the more severe the impact of these scallops on the subsequent tool path. This is critical

since subsequent tool passes for scenarios where this ratio would likely be large are between roughing and finishing mills. In addition, from the discussion of the constructive interference of the previous tool's scallops on the chip thickness and ultimately forces of the subsequent pass, it is recommended that subsequent tool passes be timed as to best avoid this outcome. While the precise indexing of the period of any cutting tool can be difficult, especially in industrial applications, the choice of the feedrate of the subsequent tool to be non-complimentary of the previous tool pass, can be sufficient to avoid periodic and spikes in force and consequently greater deflections of the subsequent tool from the desired tool path. Ultimately, with equation (62) the height of these scallops can be estimated for both the scallops developed by the trochoidal path of the cutting teeth and the scallops that result from the step over of the previous tool by simply replacing the feed per tooth parameter with the distance of the step over.

This method of simulating the machining process can be used to greatly improve sculpted surface machining operations in the aerospace, automotive, and mold and die industries just to name a few. This applies to any machining run where it can be more efficient and cost saving to run prior simulations of the machining passes than to invoke and rely upon a period of trial and error to get a rough optimization for the production of parts. In research, the model can be used a jumping off point for even more detailed and inclusive analyses of the machining process. Since the tool and workpiece are modelled separately, the model can easily be extended to allow the incorporation of tool deflection and possibly even thermal effects since the data structure is already discrete these effects can be stored numerically in the indices of the voxels that remain on the workpiece and in the indices of the voxels removed. This analysis showed that the effects of surface

roughness can easily be accounted for by the simulation, and this opens the door for the model to be used to analyse data reconstructed from precision measurement techniques like structured light and x-ray computed tomography of additively manufactured parts that require precision post processing with machining operations. Moreover, the physical part resulting from the real world machining process can be scanned and compared with the workpiece resulting from the virtual machining for improving accuracy of surface roughness predictions.

REFERENCES

[1] A. Williams, S. Barrus, R. K. Morley and P. Shirley, "An Efficient and Robust Ray-Box Intersection Algorithm," *Journal of Graphics Tools,* 2005.

[2] M. E. Merchant, "An Interpretive Look at 20th Century Research on Modeling of Machining," *Machining Science and Technology,* vol. 2, no. 2, pp. 157-163, 1998.

[3] J. R. Nicholson, "Made In America: Machinery," 2013. [Online]. Available: http://webcache.googleusercontent.com/search?q=cache:rr-YQ-2T7gwJ:www.esa.doc.gov/sites/default/files/machineryindustryprofile_0.pdf+&cd=8&hl=en&ct=clnk&gl=us. [Accessed 19 November 2018].

[4] Y. Shah, "Contribution of Machining Tools to Boost Global Economy," Yash Machine Tools, 30 March 2015. [Online]. Available: https://www.yashmachine.com/blog/contribution-of-machining-tools-to-boost-global-economy/. [Accessed 20 November 2018].

[5] M. E. Martellotti, "An Analysis of the Milling Process," *Transactions of the ASME,* vol. 63, no. 8, pp. 677-700, 1941.

[6] K. F. Ehmann, S. G. Kapoor, R. E. Devor and I. Lazoglu, "Machining Process Modelling: A Review," *Journal of Manufacturing Science and Engineering,* vol. 119, pp. 655-663, 1997.

[7] M. E. Merchant, "Mechanics of the Metal Cutting Process I: Orthogonal Cutting and the Type 2 Chip," *Journal of Applied Physics,* vol. 16, no. 5, pp. 267-275, May 1945.

[8] H. Ernst, "Physics of Metal Cutting," *American Society for Metals,* p. 24, 1938.

[9] E. J. A. Armarego and C. J. Epp, "An Investigation of Zero Helix Peripheral Up-Milling," *International Journal of Machine Tool Design,* vol. 10, pp. 273-291, 1970.

[10] P. L. B. Oxley and A. P. Hatton, "Shear Angle Solution Based on Experimental Shear Zone and Tool Chip Interface Stress Distribution," *International Journal of Mechanical Sciences,* vol. 5, no. 1, pp. 41-55, 1963.

[11] F. Koenigsberger and A. P. Sabberwal, "An Investigation into the Cutting Force Pulsations During Milling Operations," *International Journal of Machine Tool Design and Research,* vol. 1, no. 1-2, pp. 15-33, 1961.

[12] S. G. Kapoor, R. E. Devor, R. Zhu, R. Gajjela, G. Parakkal and D. Smithey, "Development of Mechanistic Models for the Prediction of Machining Performance: Model Building Methodology," *Machining Science and Technology,* vol. 2, no. 2, pp. 213-238, 2007.

[13] E. Budak, "Analytical Models for High Performance Milling. Part I: Cutting Forces, Structural Deformations and Tolerance Integrity," *International Journal of Maching Tools and Manufacture ,* vol. 46, pp. 1478-1488, 2006.

[14] Y. Altintas, Manufacturing Automation, Cambridge: The Press Syndicate of the University of Cambridge, 2000, pp. 4-47.

[15] H. J. Fu, R. E. DeVor and S. G. Kapoor, "A Mechanistiv Model for the Prediction of the Force System in Face Milling," *Journal of Engineering for Industry ,* vol. 106, no. 1, pp. 81-88, 1984.

[16] J. W. Sutherland and R. E. DeVor, "An Improved Method for Cutting Force and Surface Error Prediction in Flexible End Milling Systems," *Journal of Engineering for Industry,* vol. 108, no. 4, pp. 269-279, 1986.

[17] E. A. Armarego and R. C. Whitfield, "Computer Based Modelling of Popular Machining Operations for Force and Power Prediction," *Cirp Annals,* vol. 34, no. 1, pp. 65-69, 1985.

[18] Y. Altintas, Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design, New York: Cambridge University Press, 2012.

[19] A. Azeem, H.-Y. Feng and L. Wang, "Simplified and Efficient Calibration of a Mechanistic Cutting Force Model for Ball-End Milling," *International Journal of*

Machine Tools and Manufacture, vol. 44, no. 2004, pp. 291-298, 25 September 2003.

[20] J. Gradisek, M. Kalveram and K. Weinert, "Mechanistic Identification of Specific Force Coefficients for a General End Mill," *International Journal of Machine Tools and Manufacturing,* vol. 44, pp. 401-414, 2004.

[21] M. Wan, W.-J. Pan, W.-H. Zhang, Y.-H. Ma and Y. Yang, "A unified Instantaneous Cutting Force Model for Flat End MIlls with Variable Goemetries," *Journal of Materials Processing Technology,* vol. 214, pp. 641-650, 2014.

[22] J. J. Childs, Numerical Control Part Programming, Alexandria, Virginia: Industrial Press, 1973.

[23] L. Zhen, Y. Li and S. Y. Liang, "A Generalized Model of Milling Forces," *International Journal of Macnufacturing Technology,* vol. 14, pp. 160-171, 1997.

[24] Y. Altintas and A. Spence, "End Milling Force Algorithms for CAD Systems," *Annls of the CIRP,* vol. 40, no. 1, pp. 31-34, 1991.

[25] M. Kaymakci, Z. M. Kilic and Y. Altintas, "Unified Cutting Force Model for Turning, Boring, Drilling, and Milling Operations," *International Journal of Machine Tools and Manufacture,* Vols. 54-55, pp. 34-45, 2012.

[26] W. Ferry and D. Yip-Hoi, "Cutter-Workpiece Engagement Calculations by Parallel Slicing for Five-Axis Flank Milling of Jet Engine Impellers," *Journal of Maufacturing Science and Engineerign,* vol. 130, pp. 1-12, 2008.

[27] S. Engin and Y. Altintas, "Mechanics and Dynamics of General Milling Cutters. Part 1: Helical End Mills," *International Journal of Machine tools and Manufacture,* vol. 41, pp. 2196-2212, 2001.

[28] S. D. Merdol and Y. Altintas, "Virtual Simulation and Optimization of Milling Operations Part I: Process Simulation," *Journal of Manufacturing Science and Engineering,* vol. 130, no. 051004, pp. 1-12, 2008.

[29] G. M. Kim, "Cutting Force Prediction Of Sculpted Surface Ball-End Milling Using Z-Map," *International Journal of Machne Tools and Manufacture,* vol. 40, pp. 277-291, 2000.

[30] E. J. A. Armarego and N. P. Deshpande, "Force Prediction Models and CAD/CAM Software for Helical Tooth Milling Processes. I Basic Approach and Cutting Analyses," *International Journal of Production Research,* vol. 31, no. 8, pp. 1991-2009, 1993.

[31] H. El-Mounayri, M. A. Elbestawi, A. D. Spence and S. Bedi, "General Geometric Modelling Approach for Machining Process Simulation," *International Journal of Manufacturing Technology,* vol. 13, pp. 237-247, 1997.

[32] T. Tucker, "SculptPrint Blog," Tucker Innovations, 2018. [Online]. Available: http://sculptprint.com/blog/. [Accessed 23 November 2018].

[33] D. Jang, K. Kim and J. Jung, "Voxel-Based Virtual Multi-Axis Machining," *International Journal of Advanced Manufacturing Technology,* vol. 16, pp. 709-713, 2000.

[34] J. Tarbutton, T. Kurfess and T. Tucker, "Graphics Based Path Planning for Multi-Axis Machine Tools," *Computer Aided Design and Applications,* vol. 7, no. 6, pp. 835-845, 2010.

[35] J. Tarbutton, T. Kurfess, T. Tucker and D. Konobrytskyi, "Gouge-Free Voxel-Based Machining for Parallel Processors," *International Journal of Manufacutring Technology,* vol. 69, pp. 1941-1953, 2013.

[36] J. A. Tarbutton, T. R. Kurfess and T. M. Tucker, "Graphics Based Path Planning for Multi-Axis Machine Tools," *Computer Aided Design and Applicatoins,* vol. 7, pp. 835-845, 2008.

[37] J. Tarbutton, T. R. Kurfess, T. Tucker and D. Konobrytskyi, "Gouge-Free Voxel Based Machining for Parallel Processors," *International Journal of Advance Manufacturing Technology,* vol. 69, no. 2013, pp. 1941-1953, 13 July 2013.

[38] S. J. Wou, Y. C. Shin and H. El-Mounayri, "Ball End Milling Mechanistic Model Based on a Voxel-Based Geometric Reresentation and a Ray Casting Technique," *Journal of Manufacturing Processes,* vol. 15, 2013.

[39] J. Pawasauskas, "Volume Visualization with Ray Casting," 18 February 1997. [Online]. Available: http://web.cs.wpi.edu/~matt/courses/cs563/talks/powwie/p1/ray-cast.htm#Introduction. [Accessed 21 October 2017].

[40] B. Zalik, G. Clapworthy and C. Oblonsek, "An Efficient Code-Based Voxel-Traversing Algorithm," *Computer Graphics Forum: The Eurographics Association,* vol. 16, no. 2, pp. 119-128, 1997.

[41] J. Amanatides and A. Woo, "A Fast Voxel Traversal Algorithm for Ray Tracing," *Proceedings in Eurographics,* pp. 1-10, 1987.

[42] B. Smits, "Efficiency Issues for Ray Tracing," *Journal of Graphics Tools,* pp. 1-12, 1999.

[43] G. Yucesan and Y. Altintas, "Improved Modelling of Cutting Force Coefficients in Peripheral Milling," *International Journal of Machine Tools Manufacturing,* vol. 34, no. 4, pp. 473-487, 1994.

[44] M. A. Rubeo and T. L. Shmitz, "Milling Force Modelling: A Comparison of Two Approaches," *Pocedia Manufacturing,* vol. 5, no. 44, pp. 90-105, 2016.

[45] F. Kharfi, "Mathematics and Physics of Computed Tomography (CT): Demonstrations and Practical Examples," 2013. [Online]. Available: http://cdn.intechopen.com/pdfs/43595/InTech-Mathematics_and_physics_of_computed_tomography_ct_demonstrations_and_practical_examples.pdf. [Accessed May 2018].

[46] E. Budak, Y. Altintas and E. J. A. Armarego, "Prediction of Milling Force Coefficients form Orhtogonal Cutting Data," *Journal of Manufacturing Science and Engineering,* vol. 118, no. May, pp. 216-224, 1996.

[47] T. M. Tucker, T. Kurfess and D. Konobrtskyi, "Hybrid Dynamic Tree Data Structure and Accessibility Mapping For Computer Numerically Controlled Machining Path Planning". United States of America Patent US 9971335 B2, 15 May 2018.

[48] A. Narvekar, Y. Huang and J. H. Oliber, "Intersection of Rays with Parametric Envelope Surfaces Representing five-axis NC Milling Tool Swept Volume," *Proceedings of ASME Advances in Design Automation,* vol. 2, pp. 223-230, 1992.

[49] Tyson Tool Company Limited, "Technical-Mill-Formulas.pdf," [Online]. Available: http://www.tysontool.com/tech-mill-formulas.pdf. [Accessed 20 November 2018].

[50] J. H. Oliver and Y. Huang, "Integrated Simulation, Error Assesment, and Tool Path Correction for five-Axis NC Milling," *Journal of Manufacturing Systems,* vol. 14, no. 5, p. 331, 1995.

[51] J. G. Cleary and G. Wyvill, "Analysis of an Algorithm for Fast Ray Tracing Using Uniform Space Subdevision," *The Visual Computer,* pp. 65-83, 1988.

[52] Y. Altintas, "End Milling Force Algorithms for CAD Systems," *Annals of CIRP*, vol. 40, no. 1, pp. 31-34, 1991.

[53] S. Ding and A. N. Poo, "Oriented Bounding box and Octree Global Interference Detection in 5-axis Machining of Free=Form Surfaces," *Computer Aided Design*, vol. 36, pp. 1281-1294, 2004.